



## Help for Tips

[Properties](#)

[Events](#)

[Frequently Asked Questions](#)

### Registration Information

### Order Form

### Getting Custom Controls Written

### Licensing Information

#### **Description**

Tips VBX/OCX provides Microsoft style tool tips. Small windows pop-up with bits of info when the user pauses the mouse over your controls. No code required, just set the Tag properties. Tips uses the hWnd property when attaching itself to a control.

#### **File Name**

TIPS1.VBX, TIPS16.OCX, TIPS32.OCX

#### **ActiveX / OCX Object Name**

Mabry.TipsCtrl

#### **ActiveX Compatibility**

VB 4.0 (32-bit) and 5.0

#### **ActiveX Built With**

Microsoft Visual C++ v4

#### **ActiveX - Required DLLs**

MFC40.DLL (October 6th, 1995 or later)

OLEPRO32.DLL (October 6th, 1995 or later)

MSVCRT40.DLL (September 29th, 1995 or later)

#### **VBX Object Type**

MabryTips

#### **VBX Compatibility**

VB 2.0, 3.0 and 4.0 (16-bit)

#### **VBX Built With**

Microsoft Visual C++ v1.5

#### **Remarks**

When the Enabled property is set to True (default), Tips automatically shows tool tips.

Tips should be placed on all forms where its functionality is required (including MDI forms and children). Tips can be placed directly on an MDI form.

At design time, Tips automatically shows you what a tip will look like. Set colors (BackColor and ForeColor) and font to get the look you desire. Then, set the Tag properties of the controls for which you want tips. Tips will automatically display those Tags at run-time, when the mouse cursor pauses over the controls.

#### **Usage Note**

Tips OCX does not show a help window/tool tip for Label and Image controls. It does, however, show a tip for Picture Boxes.

**Distribution Note** When you develop and distribute an application that uses this control, you should install the control file into the user's Windows SYSTEM directory. The control file has version

---

information built into it. So, during installation, you should ensure that you are not overwriting a newer version.

---

Close

## Tips Properties

Properties that have special meaning for this control or that only apply to this control are marked with an asterisk (\*).

**Align** Property  
**BackColor** Property  
**Caption** Property  
**Enabled** Property  
**FontBold** Property  
**FontItalic** Property  
**FontName** Property  
**FontSize** Property  
**FontStrikethru** Property  
**FontUnderline** Property  
**ForeColor** Property  
**Height** Property  
**Index** Property  
\*Interval Property  
\*Interval2 Property  
**Left** Property  
**Name** Property  
**Parent** Property  
**Tag** Property  
\*TipAlignment Property  
**Top** Property  
\*UseTag Property  
\*Version Property  
**Width** Property  
\*XOffset Property  
\*YOffset Property

Close

## Tips Events

Events that have special meaning for this control or that only apply to this control are marked with an asterisk (\*).

\***MouseOver** Event

\***TipDisplay** Event

\***TipHide** Event

Close

## Frequently Asked Questions

### General Questions

[Why does the whole form re-paint when a tip goes away?](#)

[How can I get Tips to work with Image controls and other controls that don't work with Tips \(such as SCommand\)?](#)

[Why doesn't Tips work with every control?](#)

## **Why does the whole form re-paint when a tip goes away?**

### Frequently Asked Questions

This usually happens when the form's ClipControls property is set to True. This is easily fixed by changing ClipControls to False.

## **How can I get Tips to work with Image controls and other controls that don't work with Tips (such as SSCommand)?**

[Frequently Asked Questions](#)

Place another control (such as a Picture Box, SSPanel, anything that will blend in with your background and that Tips works with) behind the control that does not work. Set the Tag property of the bottom control to the desired tip.

## **Why doesn't Tips work with every control?**

### Frequently Asked Questions

Tips works by watching the mouse movement and clicks (the technique is called hooking the mouse). Programs and controls that hook the mouse are supposed to follow certain rules of behavior, the most important of which is: Be Polite.

What does this mean? When a mouse message comes in, the message is passed to every program and DLL that has hooked the mouse. Each program or DLL that has hooked the mouse is responsible for passing the message on to the next program or DLL in the list.

Some programs or DLLs don't bother to pass the message on to the next program or DLL. They just use the message and dump it.

You might say "How Rude!" You would be correct. Since Tips relies on the message, Tips cannot work where there is an impolite control dumping mouse messages.

We are currently working on ways to get around these issues.



## Registration Information

### CREDITS

Tips was written by James Shields.

### CONTACT INFORMATION

Orders, inquiries, technical support, questions, comments, etc. can be sent to [mabry@mabry.com](mailto:mabry@mabry.com) on the Internet. Our mailing address/contact information is:

Mabry Software, Inc.  
Post Office Box 31926  
Seattle, WA 98103-1926  
Sales: 1-800-99-MABRY (U.S. Only)  
Voice: 206-634-1443  
Fax: 206-632-0272 or 206-364-3196  
Web: <http://www.mabry.com>

### COST

The price of Tips (control only) is US\$25 (US\$30 for International orders). The cost of Tips and the C/C++ source code (of the control itself) is US\$55 (US\$60 for International orders).

Prices are subject to change without notice.

### DELIVERY METHODS

We can ship this software to you via air mail and/or e-mail.

**Air Mail** - you will receive disks, a printed manual, and printed receipt if you choose this delivery method. The costs are:

US\$5.00	US Priority Mail
US\$10.00	AirBorne Express 2nd Day (US deliveries only)
US\$15.00	AirBorne Express Overnight (US deliveries only)
US\$45.00	International AirBorne Express.

**E-Mail** - We can ship this package to you via e-mail. You need to have an e-mail account that can accept large file attachments (which includes CompuServe, AOL, and most Internet providers). If you choose this option, please note: a printed manual is not included. We will, however, e-mail a receipt to you.

Be sure to include your full mailing address with your order. Sometimes (on the Internet) the package cannot be e-mailed, so we are forced to send it through the normal mails.

**CompuServe E-Mail** - CompuServe members can use the software registration forum (GO SWREG) to register this package. Tips' SWREG ID number is 3482. The source code version's ID number is 3483. PLEASE NOTE: When you order through SWREG, we send the registered package to your CompuServe account (not your Internet or AOL account) within a few hours.

### ORDER / PAYMENT METHODS

You can order this software by phone, fax, e-mail, mail. For your convenience, an order form has been provided that you can print out directly from this help file.

Please note that orders must include all information that is requested on our order form. Your shipment WILL BE DELAYED if we have to contact you for additional information (such as phone number, street address, etc.).

You can pay by credit card (VISA, MasterCard, American Express), check (U.S. dollars drawn on a U.S. bank), cash, International Money Order, International Postal Order, Purchase Order (established business entities only - terms net 30), or wire transfer.

### WIRE TRANSFER INFORMATION

Here is the information you need regarding our account for a wire funds transfer:

Bank Name: SeaFirst - Stone Way Branch  
Bank Address: 3601 Stone Way North  
Seattle, WA 98103  
Bank Phone: 206-585-4951  
Account Name: Mabry Software, Inc.  
Routing Number: 12000024  
Account Number: 16311706

If you are paying with a wire transfer of funds, please add US\$12.50 to your order. This is the fee that SeaFirst Bank charges Mabry Software. Also, please ADD ANY ADDITIONAL FEES THAT YOUR BANK MAY CHARGE for wire transfer service. If you are paying with a wire transfer, we must have full payment deposited to our account before we can ship your order.

Copyright © 1994-1997 by Mabry Software, Inc.



## Tips Order Form

Use the Print Topic... command from the File menu to print this order form.

**Mail this form to:** Mabry Software, Inc.  
Post Office Box 31926  
Seattle, WA 98103-1926  
  
Phone: 206-634-1443  
Fax: 206-632-0272 or 206-364-3196  
Internet: mabry@mabry.com  
Web: www.mabry.com

Where did you get this copy of Tips?

\_\_\_\_\_

Name: \_\_\_\_\_

Ship to: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Phone: \_\_\_\_\_

Fax: \_\_\_\_\_

E-Mail: \_\_\_\_\_

MC/VISA/AMEX: \_\_\_\_\_ exp. \_\_\_\_\_

P.O. # (if any): \_\_\_\_\_ Signature \_\_\_\_\_

qty ordered \_\_\_\_\_ REGISTRATION  
\$25.00 (\$30.00 international). Check or money order in U.S. currency drawn  
on a U.S. bank. Add \$5.00 per order for shipping and handling.

qty ordered \_\_\_\_\_ SOURCE CODE AND REGISTRATION  
\$55.00 (\$60.00 international). Check or money order in U.S. currency drawn  
on a U.S. bank. Add \$5.00 per order for shipping and handling.



## TipDisplay Event Example

In this example, the tip displayed is the each control's name (except controls that begin with "Form"). To try this example, paste the code into the Declarations section of a form that contains a number of various controls and a Tips control. Press F5.

```
Sub Tips1_TipDisplay (ControlName As String, HelpContextID As Long, Tag As String)
    If Left(ControlName,4) = "Form" Then
        Tag = ""
    Else
        Tag = ControlName
    End If
End Sub
```

**See Also**

**TipDisplay** Event

**See Also**

**TipDisplay** Event

## See Also

[TipDisplay](#) Event

[XOffset](#) Property

[YOffset](#) Property

## See Also

[MouseOver](#) Event

[TipAlignment](#) Property

[TipHide](#) Event

[UseTag](#) Property



## See Also

[TipDisplay](#) Event

[UseTag](#) Property

## See Also

[TipDisplay Event](#)

[TipHide Event](#)

## See Also

[TipAlignment](#) Property

[TipDisplay](#) Event

## Interval and Interval2 Properties

[See Also](#)

[Frequently Asked Questions](#)

### Description

Determines how long the mouse must not move.

### Syntax

*object.Interval* [= *long* ]

*object.Interval2* [= *long* ]

The syntax of the **Interval** and **Interval2** properties has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Tips control.
<i>long</i>	A long integer that expresses a duration in milliseconds.

### Remarks

Interval determines how long (in milliseconds) the mouse cursor must pause over a control before Tips displays a tip.

Interval2 determines how long (in milliseconds) the mouse must pause over a second control after the tip is displayed. Typically this is set short to allow the user to move from one control to the next after the first tip is shown.

### Data Type

Integer (long)

## MouseOver Event

[See Also](#)

[Frequently Asked Questions](#)

### Description

Occurs when the mouse passes over a control.

### Syntax

**Sub** *object\_MouseOver* (*[index As Integer,] ControlName As String, HelpContextID As Long, Tag As String*)

The syntax of the **MouseOver** event has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Tips control.
<i>index</i>	An integer that identifies a control if it's in a control array.
<i>ControlName</i>	A string expression that holds the name of the control that the mouse cursor is over.
<i>HelpContextID</i>	A long integer that holds the HelpContextID property of the control tha the mouse cursor is over.
<i>Tag</i>	A string expression that, when set in this event, determines the text to display in the tool tip. If the <u>UseTag property</u> is set to True, this parameter is initially set to the UseTag property of the control specified in <i>ControlName</i> .

### Remarks

This event occurs when the mouse passes into a control. This can be used to feed a status bar.

The argument *ControlName* holds the name of the control the mouse is in.

The argument *HelpContextID* holds the HelpContextID property of the control selected. This property can be set in the properties window at design-time.

If the UseTag property is True, the argument *Tag* holds the Tag property of the control selected. Otherwise, the *Tag* argument is empty.

## TipAlignment Property

[See Also](#)

[Frequently Asked Questions](#)

### Description

Determines where the tip is displayed.

### Syntax

*object*.**TipAlignment** [= *align* ]

The syntax of the **TipAlignment** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Tips control.
<i>align</i>	An integer that determines the location of the tool tip in relation to the control or cursor.

### Remarks

This property determines where the tip is placed relative to the cursor or the control. It can be one of the following values:

<b><u>Constant</u></b>	<b><u>Value</u></b>	<b><u>Description</u></b>
msTACursor	0	Cursor
msTACursorFollow	1	Cursor (follow) - the tip follows the cursor
msTATopLeft	2	Upper left corner of control
msTATopCenter	3	Upper center of control
msTATopRight	4	Upper right corner of control
msTAMiddleLeft	5	Left side of control
msTACenter	6	Center of control
msTAMiddleRight	7	Right side of control
msTABottomLeft	8	Lower left corner of control
msTABottomCenter	9	Lower center of control
msTABottomRight	10	Lower right corner of control

### Data Type

Integer (enumerated)

## TipDisplay Event

[See Also](#)

[TipDisplay Event Example](#)

[Frequently Asked Questions](#)

### Description

Occurs when Tips is about to display a tip.

### Syntax

**Sub** *object*\_**TipDisplay**([*index* **As Integer**,] *ControlName* **As String**, *HelpContextID* **As Long**, *Tag* **As String**)

The syntax of the **TipDisplay** event has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Tips control.
<i>index</i>	An integer that identifies a control if it's in a control array.
<i>ControlName</i>	A string expression that holds the name of the control that the mouse cursor is over.
<i>HelpContextID</i>	A long integer that holds the HelpContextID property of the control tha the mouse cursor is over.
<i>Tag</i>	A string expression that, when set in this event, determines the text to display in the tool tip. If the <a href="#">UseTag property</a> is set to True, this parameter is initially set to the UseTag property of the control specified in <i>ControlName</i> .

### Remarks

This event only occurs if the Enabled property is set to True (default). Also, this event only fires for the form that the control is on. If there are multiple forms that need tool tips, you must place Tips on each one.

The argument *ControlName* holds the name of the control the mouse has paused over.

The argument *HelpContextID* holds the HelpContextID property of the control selected. This property can be set in the properties window at design-time.

If the [UseTag property](#) is True, the argument *Tag* holds the Tag property of the control selected.

Otherwise, the Tag argument is empty. This is the text that Tips is about to display. You can change it to something else, if needed. If you set it to an empty string, the tip window is not displayed.

## TipHide Event

[See Also](#)

[Frequently Asked Questions](#)

### Description

Occurs when Tips removes the tip window from the screen.

### Syntax

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

The syntax of the **TipHide** event has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Tips control.
<i>index</i>	An integer that identifies a control if it's in a control array.

### Remarks

This event occurs when Tips removes (hides) the tip window from the screen.



## UseTag Property

[See Also](#)

[Frequently Asked Questions](#)

### Description

Determines whether or not to use the Tag property as the default.

### Syntax

*object*.**UseTag** [= *boolean* ]

The syntax of the **UseTag** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Tips control.
<i>boolean</i>	A boolean value that determines if a control's Tag property is used as the default tool tip.

### Remarks

If this property is True, Tips uses the Tag property of the control the mouse is over as the default value to display. If this value is False, Tips uses no default. You may want to set this property to False if you are using other controls that require values in the Tag property.

This property defaults to True.

### Data Type

Integer (boolean)

## Version Property

Frequently Asked Questions

### Description

Shows the version of the control.

### Syntax

*object*.**Version**

The syntax of the **Version** property has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Tips control.

### Remarks

This property holds the current version of the control. It is read-only and available at both design-time and run-time.

### Data Type

String

## XOffset and YOffset Properties

[See Also](#)

[Frequently Asked Questions](#)

### Description

Determines where to place the tip at run-time.

### Syntax

*object.XOffset* [= *integer* ]

*object.YOffset* [= *integer* ]

The syntax of the **XOffset** and **YOffset** properties has these parts:

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	A Tips control.
<i>integer</i>	An integer that holds the distance, in pixels, to place the tool tip from the cursor. Positive values move the tool tip down and right (on the screen), negative values are up and left.

### Remarks

These properties determine where to place the tip when it's displayed at run-time. These values are in pixels. The tip is placed relative to the hot-spot of the mouse cursor.

### Data Type

Integer

## **Getting Custom Controls Written**

If you or your organization would like to have custom controls written, you can contact us at the following:

Mabry Software, Inc.  
Post Office Box 31926  
Seattle, WA 98103-1926  
Phone: 206-634-1443  
Fax: 206-632-0272 or 206-364-3196  
Internet: [mabry@mabry.com](mailto:mabry@mabry.com)

You can also contact Zane Thomas. He can be reached at:

Zane Thomas  
Post Office Box 121  
Indianola, WA 98342  
Internet: [zane@mabry.com](mailto:zane@mabry.com)

## **Licensing Information**

### **Legalese Version**

Mabry Software grants a license to use the enclosed software to the original purchaser. Copies may be made for back-up purposes only. Copies made for any other purpose are expressly prohibited, and adherence to this requirement is the sole responsibility of the purchaser.

Customer written executable applications containing embedded Mabry products may be freely distributed, without royalty payments to Mabry Software, provided that such distributed Mabry product is bound into these applications in such a way so as to prohibit separate use in design mode, and that such Mabry product is distributed only in conjunction with the customers own software product. The Mabry Software product may not be distributed by itself in any form.

Neither source code for Mabry Software products nor modified source code for Mabry Software products may be distributed under any circumstances, nor may you distribute .OBJ, .LIB, etc. files that contain our routines. This control may be used as a constituent control only if the compound control thus created is distributed with and as an integral part of an application. Permission to use this control as a constituent control does not grant a right to distribute the license (LIC) file or any other file other than the control executable itself. This license may be transferred to a third party only if all existing copies of the software and its documentation are also transferred.

This product is licensed for use by only one developer at a time. Mabry Software expressly prohibits installing this product on more than one computer if there is any chance that both copies will be used simultaneously. This restriction also extends to installation on a network server, if more than one workstation will be accessing the product. All developers working on a project which includes a Mabry Software product, even though not working directly with the Mabry product, are required to purchase a license for that Mabry product.

This software is provided as is. Mabry Software makes no warranty, expressed or implied, with regard to the software. All implied warranties, including the warranties of merchantability and fitness for a particular use, are hereby excluded.

MABRY SOFTWARE'S LIABILITY IS LIMITED TO THE PURCHASE PRICE. Under no circumstances shall Mabry Software or the authors of this product be liable for any incidental or consequential damages, nor for any damages in excess of the original purchase price.

To be eligible for free technical support by telephone, the Internet, CompuServe, etc. and to ensure that you are notified of any future updates, please complete the enclosed registration card and return it to Mabry Software.

### **English Version**

We require that you purchase one copy of a control per developer on a project. If this is met, you may distribute the control with your application royalty free. You may never distribute the LIC file. You may not change the product in any way that removes or changes the requirement of a license file.

We encourage the use of our controls as constituent controls when the compound controls you create are an integral part of your application. But we don't allow distribution of our controls as constituents of other controls when the compound control is not part of an application. The reason we need to have this restriction is that without it someone might decide to use our control as a constituent, add some trivial (or even non-trivial) enhancements and then sell the compound control. Obviously there would be little difference between that and just plain reselling our control.

If you have purchased the source code, you may not re-distribute the source code either (nor may you copy it into your own project). Mabry Software retains the copyright to the source code.

Your license is transferable. The original purchaser of the product must make the transfer request. Contact us for further information.

The sample versions of our products are intended for evaluation purposes only. You may not use the sample version to develop completed applications.

