

# TOOL TIPS FOR MICROSOFT ACCESS 2.0

Version 3.0

December 12, 1994

by Dave Schlosser

CompuServe ID: 73631,646

with

Cue Cards and Enhancements by

Chris Patterson

CompuServe ID: 72713,2136

## Features of this Tool Tip Implementation.

1. Add tool tips to your controls
2. Refer to cue cards (for reminders such as argument order, which events to add tool tip routines to, function names, etc.)
3. Specify that a character be underlined, so tool tips will reflect the ALT-<letter> access key specified for a control
4. Change the status bar text as the mouse passes over controls.
5. Enable/disable tool tips
6. Enable/disable status bar text modification
7. Change the tool tip display delay time
8. Requires Access 2.0 or above to run.

## Files

This zip file contains two Microsoft Access library databases, 1) TTIPCUE.MDB, which contains cue cards and the directions for using this tool, and 2) TOOLTIP.MDA which should be shipped with your applications. The tool tip library (TOOLTIP.MDA) contains all of the code needed to implement floating tool tips (ala toolbars) on buttons or any control you desire. Directions for adding tool tip cue cards to your utility toolbar are also included in this document.

## To View an Example and the Cue Cards:

1. You must first add TOOLTIP.MDA [Libraries] section of your MSACC20.INI file. Save the file and restart Access. Example:

```
[Libraries]
...
c:\access\library\tooltip.mda=ro
```

-OR- Use the MS Access Add-in Manager to install TOOLTIP.MDA.

2. Open the TTIPCUE.MDB data base. The main cue card is automatically displayed. Two of the last three options are examples that utilize Tool Tips. Select one or both of these examples and then place the cursor over any of the controls and wait half a second the tool tip to appear. These two examples (forms) may be viewed in design mode and are called yfrmToolTipsExampleMenu and yfrmToolTipsExampleForm, respectively.

## Installation

In order to use the tool tip routines, TOOLTIP.MDA has to be installed in the [Libraries] section of your MSACC20.INI file. In order to use the cue cards from your utility toolbar, TTIPCUE.MDB should also be added to your MSACC20.INI file. Save the file and restart Access. Example:

```
[Libraries]
...
c:\access\library\tooltip.mda=ro
c:\access\library\TTIPCUE.MDB=ro
```

-OR- Use the MS Access Add-in Manager to install TOOLTIP.MDA.

## Connecting Tool Tip Cue Cards to the Utility Toolbar

The utility 1 and utility 2 toolbars are used by all databases, making them ideal for the developer to customize them to their needs.

To add tool tip cue cards to your utility bar do the following:

1. Import into your current database the macro, "mcrUtilityToolBars" from TTIPCUE.MDB. This macro file contains two macros, one to open the Add [Tool Tips] cue card and one to open the Main cue card. I believe most developer's will want to call up the Add cue card first instead of the Main cue card. There is a main menu button on each cue card.
2. Add to the utility bar the Add (or if you like, the Main) cue card macro. For detailed instructions, the Access manual, Appendix B, Creating Your Own Buttons, page 725. Or search on-line help under "Toolbar Buttons, Creating" and then topic "Creating a Toolbar Button that runs a Macro".

This completes the addition of Tool Tip Cue Cards to the Utility Toolbar. But do remember that for each new database you create you will need to copy the macro "mcrUtilityToolBars" into it if you want to view the cue cards.

## Implementing Tool Tips

### 1. Controls Needing Tool Tips

On all controls where you want tool tips shown, you'll need to write a **MouseMove** event handler. The event handler should call `ttShowTip` to tell the tool tip window what description should be displayed. Example:

```
=ttfShowTip ("&Projects", [Projects Button])
```

The first parameter is the text to show in the tool tip. The "&" will cause the "P" to be underlined indicating that ALT-P is the same as clicking on this control. Don't forget you need to add an "&" before a "P" in the caption of the control itself. The second parameter is the control itself. The tool tip library displays the control's **StatusBarText** property on the status bar as the cursor passes over the control. It also uses the control's **Width** and **Height** properties to set the sensitivity.

Two alternative methods to show tool tips are:

```
=ttfShowTipCaption ([Projects Button])
=ttfShowTipTag ([Projects Button])
```

The first example uses the **Caption** property of the control for the tool tip text. The second uses the **Tag** property for the text.

If you need to include other code in the **MouseMove** event, then call the routines as subroutines with "tt" instead of "tff". In other words,

```
ttShowTip "&Projects", Me![Projects Button]
ttShowTipCaption Me![Projects Button]
ttShowTipTag Me![Projects Button]
```

Somehow the tool tip needs to be turned off when action is taken on the control, such as a button click or the user begins typing. For buttons, set up the **Click** event handler to call the subroutine `HideTip`. Example:

```
Sub Button_Click()
    ttHideTip
    ... do something...
End Sub
```

For events where the `HideTip` action is the only action (Form `OnDeactivate`), then use `=tffHideTip()` on the event property. Using functions over subroutines is preferred because when a control is copied so are the function calls (event procedures are not).

## 2. Other Controls on the Form

Technically, other controls on the form not using tool tips do not need to be modified. The tool tip routines detect when the mouse cursor moves away from the originating control. However, you can force the tool tip to disappear by coding a **MouseMove** event handler as follows:

```
Sub NoTip_MouseMove(Button As Integer, Shift As Integer,
    X As Single, Y As Single)
    ttHideTip
    ...do something....
End Sub
```

similar routine can be coded for the form or section **MouseMove** event as well.

## 3. Inspecting and Setting Delay Time

The delay time before appearing can be set. To set the delay time:

```
i = ttSelectTipDelay(500)
```

This sets the delay time to 500 milliseconds (1/2 second). The variable `i` will contain the old delay value. To inspect just the value of the delay without setting it, execute:

```
i = ttSelectTipDelay(Null)
```

Setting the delay value to less than 75 milliseconds can cause unpredictable results.

## 4. Turning Tips On and Off

Tool tips can be enabled and disabled. This setting does not affect showing the control's **StatusBarText** on the status bar. By default, tips are always enabled. To enable tool tips:

```
i = ttSelectTipState(True)
```

To disable tool tips, set `True` to `False`. The variable `i` will contain the old state. To inspect just the value of the state without setting it, execute:

```
i = ttSelectTipState(Null)
```

## 5. Turning Status Bar On and Off

Showing the status bar text can be enabled and disabled. By default, status bar prompts are always enabled. To disable status bar prompts:

```
i = ttSelectBarState(False)
```

To enable status bar again, set False to True. The variable `i` will contain the old state. To inspect just the value of the state without setting it, execute:

```
i = ttSelectBarState(Null)
```

## 6. Unloading the Tip Window

The tool tip window can be closed and unloaded completely. If you want to make the window disappear, `HideTip` instead. This routine will cause the form to be completely loaded from disk the next time it is needed. Syntax:

```
i = ttCloseTip()
```

## 7. `ttMoveTip` Subroutine

This subroutine performs all of the work to display the tip window. It is called when the delay timer expires. Calling this routine directly can create some unpredictable results.

## 8. `ttInit` Subroutine

The tool tip add-in is programmed to automatically initialize itself the first time it is used. This can cause a delay in opening the first tool tip. If you wish, call the initialize subroutine in the form's **OnLoad** event:

```
ttInit
```

## Revisions

### 3.0 Version      December 12, 1994

Added "ttf" functions to permit direct calls from event properties.  
Added code so tips display sooner if already visible.  
Added code to detect active window so tips do nothing if another app has focus.  
Fixed error where GDI-Resources were not being freed properly.  
Updated Cue Cards to reflect new functionality.

### 2.2 Version      October 19, 1994

Errors in computing the width of the window were fixed once and for all (hopefully).  
Added `ttShowTipCaption` and `ttShowTipTag` routines. Made `ttInitTip` public.

### 2.1 Version      October 17, 1994

Incorporated Chris's revisions with Dave's and thought the scaling problem was solved.  
Converted to an ADD-IN.

### 2.0 Version

Chris's cue cards were added, tip box changed from a text box to a label, and routine names were all prefixed with "tt". Minor bugs fixed.

### 1.0 Version

Original version.