

TBMsgBox - An Enhanced Message Box

TBMsgBox expands upon the functions provided by Visual Basic's MsgBox. Features include:

- Exactly duplicates VB 4.0 MsgBox functions.
- Allows for additional, more meaningful user response by allowing the command captions to be more descriptive.
- Can be used to provide bilingual messages, with corresponding captions.
- Can display alternative and user specified icons.
- Can be used to create "Splash Screens".
- Can contain a "Help" button or no buttons.
- A Wizard which creates TBMsgBox or Visual Basic code.
- Supports Windows 95 and Windows NT.

Table of Contents

Click one of the following:

[TBMsgBox Concepts and Facilities](#)

[The MsgBox Wizard](#)

[Registration and Ordering TBMsgBox](#)



Inc.

913 Hacienda Way
Millbrae, CA 94030
CompuServe: 71662,205
415-737-0870 Fax: 415-737-0871
Copyright 1996

Last updated: February 26, 1996

File: g:\tbmsgbox\help\tbmsgbox.doc

Version: 4.01

Version Date: March 1996

TBMsgBox Concepts

Click one of the following topics:

[TBMsgBox Concepts](#)

[Sample VB and TBMsgBox Messages](#)

[TBMsgBox Components](#)

[TBMsgBox.Frm and TBMsgBox.Bas](#)

[Calling TBMsgBox](#)

[Message Alignment and Fonts](#)

[Command Captions and Fonts](#)

[Alternate Icons](#)

[Selecting An Alternate Icon](#)

[Alternate Icons - Technical Information](#)

[Saving and Restoring Globals](#)

[Ctrl-P Prints the Message](#)

[Message Location](#)

[Splash Screens](#)

[Help Commands](#)

[Mouse "Snap-To" Default Button](#)

[No Buttons](#)

[No Title](#)

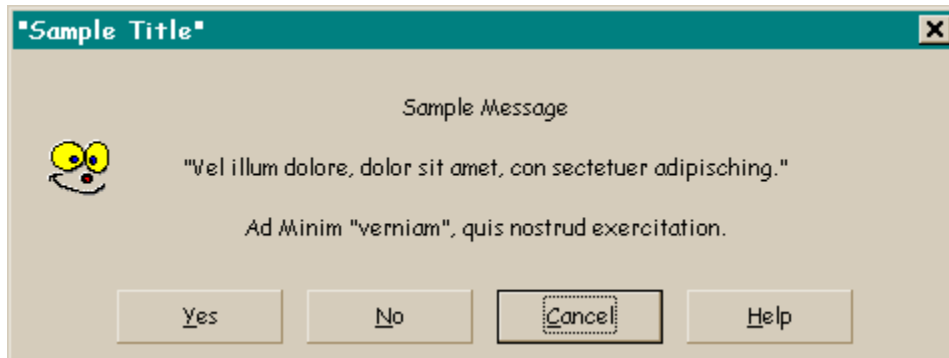
[VGA Sized Messages](#)

[Hide Rather Than Unload](#)

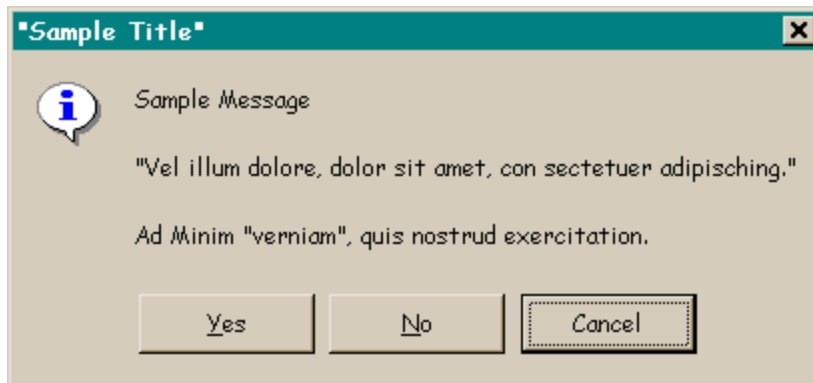
[MDI Forms.Count Routines](#)

[Converting from VB MsgBox to TBMsgBox](#)

Sample TMsgBox and VB 4.0 Messages



TMsgBox - Version 4.00



VB Message - Version 4.00

TBMsgBox Components

System Components

<u>TBMsgBox.Frm</u>	Message Box Form
TBMsgBox.Frx	Binary Data
TBMsgBox.Bas	Subroutines, Functions and Globals

Message Wizard Components

<u>TBMsgWiz.Exe</u>	Message "Wizard"
TBMsgWiz.Vbp	VB project file, used to run the system
TBMsgWiz.Frm/Frx	Demo and Generate TBMsgBox messages
TBMsgCap.Frm/Frx	Command captions and fonts
TBMsgFnt.Frm/Frx	Message alignment and fonts
TBMsgHlp.Frm/Frx	Add a Help Button
TBMsgIco.Frm/Frx	Alternate icons
TBMsgMsg.Frm	Edit the Message
TBMsgNew.Frm/Frx	Display What's New
TBMsgNew.Bas	What's New Messages
TBMsgWiz.Bas	Subroutines, etc.
TBMsgIni.Bas	Save Settings Subroutines
TBMsgHst.Bas	History of TBMsgBox Enhancements
TBMsgBox.Hlp	This Help File
MsgBSubs.Hlp	MsgBox Subs and Functions
TBMsgBox.Ghc	Help System Constants
TBMsgDef.Ghc	

Not Distributed

<u>VB40032.DLL</u>	VB 32-bit Runtime Support
<u>VB40016.DLL</u>	VB 16-bit Runtime Support
<u>ComDlg32.OCX</u>	Used for font selection
<u>ThreeD32.OCX</u>	Used for 3D Command effect

VB40032.Dll or VB40016.Dll is used by TBMsgBox. ThreeD32.OCS and ComDlg32.OCX are not used by TBMsgBox, they are only used by TBMsgWiz.

TBMsgBox.Frm and TBMsgBox.Bas

You need to include TBMsgBox.Frm and TBMsgBox.Bas in your Visual Basic project to obtain TBMsgBox support. Use VB's File, Add menu option to add the TBMsgBox form and module.

Calling Sequence

The TBMsgBox calling sequence is identical to the VB MsgBox calling sequence. See [Calling TBMsgBox](#) for details.

Supplied Arguments

Both TBMsgBox and VB's MsgBox accepts one to five arguments. See [Converting from VB MsgBox to TBMsgBox](#).

Special Properties

You can change the default message alignment and font properties by changing global TBMsgBox variables. See: [Message Alignment and Fonts](#) for details.

You can also change the default command captions and font properties. See: [Command Captions and Fonts](#).

Alternate Icons

You can use TBMsgBox to display a number of additional icons, beyond VB's basic four. See [Alternate Icons](#) for details.

Help Button - No Buttons

You can include a "Help" button with the TBMsgBox form. See: [Help Commands](#) for details. You can display the message with no buttons. See: [No Buttons](#) for details.

Splash Screens

You can use TBMsgBox to create "Splash Screens." See: [Splash Screens](#) for details.

No Titles

You can display a message without a title. See: [No Title](#) for details.

Calling TMsgBox

TMsgBox displays a message in a dialog box and waits for the user to select an option. When called as a function it returns a value indicating which option the user selected.

Function Syntax

return = TMsgBox(*prompt*[, *buttons*][, *title*][, *helpfile*, *context*])

Sample:

```
iRtn = TMsgBox ("Any Message", vbYesNo, "Any Title", _ "AnyHelp.Hlp", 1234)
```

Subroutine Syntax

TMsgBox *prompt*[, *buttons*][, *title*][, *helpfile*, *context*]

Sample:

```
TMsgBox "Any Message", vbInformation, "Any Title", "AnyHelp.Hlp", 1234
```

Named Arguments





PartDescription

return Return code supplied by TMsgBox when used as a function. Integer data type.

The TMsgBox function and statement both have these parts:

prompt String expression displayed as the message in the dialog box. String data type.

buttons Optional numeric expression that is the sum of values specifying the number and type of buttons to display, the icon style to use, the identity of the default button, and the modality. Integer data type. The following table illustrates the values used and the meaning of each group of values:

Constant	Value	Meaning
vbOKOnly	0	Display OK button only.
vbOKCancel	1	Display OK and Cancel buttons.
vbAbortRetryIgnore	2	Display Abort, Retry, and Ignore buttons.
vbYesNoCancel	3	Display Yes, No, and Cancel buttons.
vbYesNo	4	Display Yes and No buttons.
vbRetryCancel	5	Display Retry and Cancel buttons.
vbCritical	16	
vbQuestion	32	
vbExclamation	48	
vbInformation	64	
vbDefaultButton1	0	First button is default.
vbDefaultButton2	256	Second button is default.
vbDefaultButton3	512	Third button is default.
vbApplicationModal	0	Application modal. The user must respond to the message box before continuing work in the current application.
vbSystemModal	4096	System modal. The user must respond to the message box before continuing work in any application. Not valid in 32-bit applications and is ignored.

The first group of values (0-5) describes the number and type of buttons displayed in the dialog box; the second group (16, 32, 48, 64) describes the icon style; the third group (0, 256, 512) determines which button is the default; and the fourth group (0, 4096) determines the modality of

the message box. When adding numbers to create a final value for the argument buttons, use only one number from each group.

Note These constants are specified by Visual Basic for applications. As a result, the names can be used anywhere in your code in place of the actual values.

Title String expression displayed in the title bar of the dialog box. If you omit the argument title, TBMsgBox uses 'App.Title' and if none uses 'App.EXENAME'. Display a blank title by setting *title* to a space. String data type.

You can eliminate the title bar by setting MB3D_NoTitle to true before calling TBMsgBox. See [No Title](#) for details.

helpfile String expression that identifies the Help file to use to provide context-sensitive Help for the dialog box. If helpfile is provided, *context* must also be provided.

context Numeric expression that is the Help context number the Help author assigned to the appropriate Help topic. If *context* is provided, *helpfile* must also be provided.

If supplied, TBMsgBox moves the following values to its variables then continues processing the function or subroutine:

MB3D_HelpCommand = cdlHelpContext

MB3D_HelpFile = *helpfile*

MB3D_HelpData = *context*

See: [Help Commands](#) for details.

Displaying the Message

TBMsgBox displays as many characters as will fit in a LabelBox. Longer messages are truncated.

TBMsgBox breaks lines automatically at the right edge of the dialog box. If you want to set line breaks yourself, place a carriage return [ANSI character 13, vbCr] or a linefeed [ANSI 10, vbLf] before the first character of the text that is to begin each new line.

Returned Value

The TBMsgBox function *return* value indicates which button has been selected, as shown in the following table:

Constant	Value	Meaning
vbOK	1	OK button selected.
vbCancel	2	Cancel button selected.
vbAbort	3	Abort button selected.
vbRetry	4	Retry button selected.
vbIgnore	5	Ignore button selected.
vbYes	6	Yes button selected.
vbNo	7	No button selected.

If the dialog box displays a Cancel button, pressing the Esc key has the same effect as choosing Cancel.

Message Alignment and Fonts

Note: This section only applies to TBMsgBox while operating in a 16-bit environment. TBMsgBox always uses the message font found in non-client metrics while operating in the 32-bit environment.

Message alignment and font selection information can be passed to TBMsgBox in the following global variables:

```
Global MB3D_Alignment As Integer
Global MB3D_FontName As String
Global MB3D_FontSize As Single
Global MB3D_FontBold As Integer
Global MB3D_FontItalic As Integer
```

Move information into the variables before calling TBMsgBox. The variables are never changed by TBMsgBox. This means that you need only set the variables once, in the Form_Load procedure, to have the settings retained for the entire time your application is active.

A sample calling sequence which causes a centered, bold, italic, 12 point, "Arial" message to be displayed is:

```
.....
MB3D_Alignment = vbCenter
MB3D_FontName = "Arial"
MB3D_FontSize = 12
MB3D_FontBold = True
MB3D_FontItalic = True
.....
TBMsgBox "A Message", vbCritical..., "A Title"
.....
```

Defaults

The global variables are initialized with their "null" values. If you don't change them then TBMsgBox uses the "design time" alignment and font properties ["MS Sans Serif", bold, non-italic, 8.25 points with left alignment].

Constants

Text alignment values are defined in VB's help file. They are:

Alignment Constants:
vbLeftJustify
vbRightJustify
vbCenter

Command Captions and Fonts

Note: This section concerning fonts only applies to TBMsgBox while operating in a 16-bit environment. TBMsgBox always uses the menu font found in non-client metrics while operating in the 32-bit environment.

Command captions and font selection information can be passed to TBMsgBox in the following global variables:

User defined command captions:

```
Global MB3D_UD_OK As String
Global MB3D_UD_Cancel As String
Global MB3D_UD_Abort As String
Global MB3D_UD_Retry As String
Global MB3D_UD_Ignore As String
Global MB3D_UD_Yes As String
Global MB3D_UD_No As String
```

Command font properties:

```
Global MB3D_Commands_FontName As String
Global MB3D_Commands_FontSize As Single
Global MB3D_Commands_FontBold As Integer
Global MB3D_Commands_FontItalic As Integer
```

Move information into the variables before calling TBMsgBox. The variables are never changed by TBMsgBox. This means that you need only set the variables once, in the Form_Load procedure, to have the settings retained for the entire time your application is active.

A sample calling sequence which causes a non-bold, italic, 12 point, "Arial", command called "Next Page" to be displayed is:

```
.....
MB3D_UD_OK = "&Next Page"
MB3D_Commands_FontName = "Arial"
MB3D_Commands_FontSize = 12
MB3D_Commands_FontItalic = True
.....
TBMsgBoxSub "A Message", MB_Icon..., "A Title"
.....
```

Access Key

You can indicate an access key while defining the command's caption. Precede the access key with an ampersand [&], for example: "E&xit". The user selects the command by pressing the Alt-AccessKey combination, in this example: Alt-x.

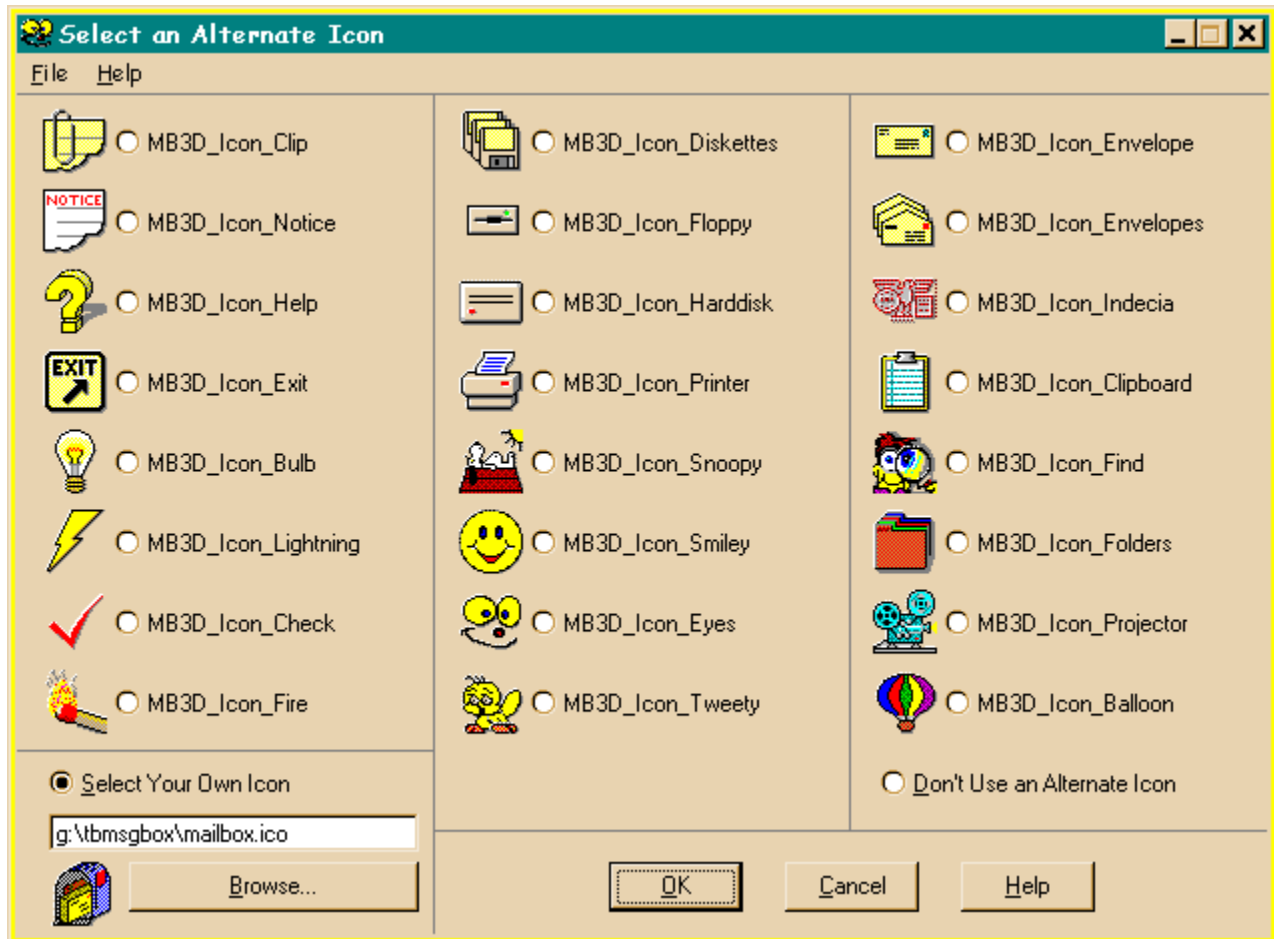
Beginning with TBMsgBox version 2.61, February 1995, users can press just the access key. Pressing the Alt key is no longer required. The key is ignored if more than one command has the same access key.

Defaults

The global variables are initialized with their "null" values. If you don't change them then TBMsgBox uses the "design time" captions and font properties ["MS Sans Serif", bold, non-italic, 8.25 points].

Alternate Icons

TBMsgBox will display 24 alternate icons in addition to the 4 standard VB icons. And TBMsgBox will also use an icon specified by the icon's fully qualified file name. The 24 alternate icons are displayed here. There are no [hot spots](#) on this screen.



Related Topic: [Selecting An Alternate Icon](#)

Selecting An Alternate Icon

Before calling TBMsgBox move the [alternate icon's index](#) to global variable **MB3D_Icon_Which**. To cause the Check Mark icon to be displayed with your message use the following code:

```
MB3D_Icon_Which = MB3D_Icon_Check
```

Revert to the use of standard VB icon's by setting **MB3D_Icon_Which** to zero. For example:

```
MB3D_Icon_Which = 0
```

Or you could use the [Global Save and Restore](#) subroutines to preserve and restore the settings.

Specifying an Icon

Move the fully qualified file name to global variable **MB3D_Icon_Filename** to cause TBMsgBox to display the specified icon with your message. For example:

```
MB3D_Icon_FileName = "d:\TBMsgBox\mailbox.ico"
```

If TBMsgBox fails to load the icon, for any reason, processing continues as though no icon file name had been specified.

Move spaces to this variable to revert to VB or alternate icon processing.

Alternate Icons - Technical Information

Icon Selection Precedence - TBMsgBox first tries to load the icon named in **MB3D_Icon_FileName**. It then uses the alternate icon specified in **MB3D_Icon_Which**. Lastly standard VB icons are used, if specified. A setting in one of the lower precedence variables is ignored if a higher precedence variable is used.

Image Controls - Alternate icons are stored in VB image controls [ImageOther()] on the TBMsgBox form [formTBMsgBox]. The image controls are indexed. The index number directly corresponds to a constant defined in TBMsgBox.Bas. The value found in **MB3D_Icon_Which** is used to show the appropriate icon using a statement similar to:

```
ImageOther (MB3D_Icon_Which).Visible = True
```

You could remove all but one of the images. Or you could add additional images. The image must have an index greater than zero because zero is used to indicate no alternate icon is to be used.

Performance Issues - Prior to version 2.4, TBMsgBox was loaded and unloaded each time it was used. If you still load rather than hide TBMsgBox each time it is used you may find the extra loading time for the alternate icons to be unacceptable. In that case, delete any unwanted icon images, leaving at least one.

If you specify an icon to be loaded, in **MB3D_Icon_FileName**, TBMsgBox loads the VB image control [ImageUserLoaded] with the icon each time TBMsgBox is activated. Statements similar to these are used:

```
ImageUserLoaded.Picture = LoadPicture (MB3D_Icon_FileName)  
ImageUserLoaded.Visible = True
```

Saving and Restoring Globals

TBMsgBox subroutines:

```
MB3D_Globals_Save  
MB3D_Globals_Restore
```

can be used to save and restore all the [user definable global variables](#).

Sample code:

```
Dim SaveArray() As Variant      'variables saved here  
MB3D_Globals_Save SaveArray()  'save settings in SaveArray  
  
TBMsgBox Message, MB_Options..., Title  
  
MB3D_Globals_Restore SaveArray() 'restore settings
```

It is much easier to use this "push-pop" code than to save individual variables then restore them after calling TBMsgBox. Be sure to dimension the array before calling the save subroutine.

Ctrl-P Prints the Message

Press **Ctrl-P** to send a TBMsgBox message to the default Windows printer. This feature is designed to aid application problem resolution.

For example: Your application issues an error message when a data access error occurs. The message displays the failing SQL statement. Your user is having trouble reading the message to you over the phone. You tell the user to press Ctrl-P, print the message then fax it to you.

Technical

Before printing, the form's backcolor is changed to white and a border is drawn around the form. TBMsgBox uses Visual Basic's **PrintForm** method. PrintForm does not print the Title or the Command Buttons.

History

This feature was implemented in TBMsgBox version 3.10, June 1995.

Message Location

By default the TBMsgBox form is centered in the screen. You can cause the message to be centered in any region of the screen. For example:

1. In an Multiple-Document Interface [MDI] application you could make messages appear centered in the MDI form.
2. In a standard or MDI application the message could appear centered in any form, Standard, Child or MDI.
3. You can "force" the message to any screen edge or corner.

MB3D_Zone Subroutine

You specify the message boundaries using one of three methods. Use subroutine MB3D_Zone to set the boundaries based on the location of a form, MB3D_Zone_FullScreen to center the message in the center of the screen, or specify each zone boundary separately. An MB3D_Zone subroutine example:

```
MB3D_Zone Me
TBMsgBoxSub "Msg", ...
```

MB3D_Zone uses one parameter - the name of a form. Another example:

```
MB3D_Zone MyMDIForm
TBMsgBoxSub "Msg", ...
```

The MB3D_Zone subroutine:

```
Sub MB3D_Zone (TheForm As Form)
MB3D_ZoneLeft = TheForm.Left
MB3D_ZoneTop = TheForm.Top
MB3D_ZoneWidth = TheForm.Width
MB3D_ZoneHeight = TheForm.Height
End Sub
```

MB3D_Zone sets four TBMsgBox global variables.

MB3D_Zone Variables

The variables are:

```
Global MB3D_ZoneLeft As Long '(27) 'left
Global MB3D_ZoneTop As Long '(28) 'top
Global MB3D_ZoneWidth As Long '(29) 'width
Global MB3D_ZoneHeight As Long '(30) 'height
```

The variables are saved and restored with [MB3D_Globals_Save/Restore](#).

MB3D_Zone_FullScreen

Use this subroutine to center a form in the middle of the screen. The subroutine:

```
Sub MB3D_Zone_FullScreen
MB3D_ZoneWidth = 0
MB3D_ZoneHeight = 0
End Sub
```

A sample:

```
MB3D_Zone_FullScreen
TBMsgBoxSub "message", ...
```

This subroutine was added in version 2.88, May 1995.

Custom Locations

Although MB3D_Zone sets these variables, you can set them to create a custom message location. Some examples:

Message in upper left corner

```
MB3D_ZoneLeft = 0
MB3D_ZoneTop = 0
MB3D_ZoneWidth = 1
MB3D_ZoneHeight = 1
```

Message in lower right corner

```
MB3D_ZoneLeft = Screen.Width
MB3D_ZoneTop = Screen.Height
MB3D_ZoneWidth = 1
MB3D_ZoneHeight = 1
```

Message at center right edge

```
MB3D_ZoneLeft = Screen.Width
MB3D_ZoneTop = Screen.Height \ 2
MB3D_ZoneWidth = 1
MB3D_ZoneHeight = 1
```

MB3D_CenterForm

This routine is used to center the TBMsgBox form. If either MB3D_ZoneWidth or MB3D_ZoneHeight is not zero then the zone boundaries are used.

```
Sub MB3D_CenterForm (TheForm As Form, OffsetLeft As Integer, OffsetTop As Integer)
'do only if Normal, not minimized or maximized
If TheForm.WindowState <> 0 Then Exit Sub

Dim FLeft As Long, FTop As Long

'any values here then use boundaries
If MB3D_ZoneWidth + MB3D_ZoneHeight > 0 Then
    FLeft = (MB3D_ZoneLeft + (MB3D_ZoneWidth \ 2)) - (TheForm.Width \ 2) - OffsetLeft
    FTop = (MB3D_ZoneTop + (MB3D_ZoneHeight \ 2)) - (TheForm.Height \ 2) + OffsetTop
    If FLeft + TheForm.Width > Screen.Width Then 'make sure it is on the screen
        FLeft = Screen.Width - TheForm.Width
    End If
    If FTop + TheForm.Height > Screen.Height Then
        FTop = Screen.Height - TheForm.Height
    End If

    If FLeft < 0 Then 'make sure it is on the screen
        FLeft = 0
    End If
    If FTop < 0 Then
        FTop = 0
    End If

Else 'center in middle of the screen
    FLeft = ((Screen.Width - TheForm.Width) \ 2) + OffsetLeft
    FTop = (((Screen.Height - TheForm.Height) \ 2) + OffsetTop) * .85
End If

'no move needed
If TheForm.Left = FLeft And TheForm.Top = FTop Then Exit Sub

TheForm.Move FLeft, FTop 'move to centered location

End Sub
```

Center Any Form in a Zone

You can use subroutine MB3D_CenterForm to center any form. For example:

In the main form:

```
MB3D_Zone Me           'set the zone  
SecondaryForm.Show     'load the form
```

In SecondaryForm's Form_Load:

```
MB3D_CenterForm Me, 0, 0 'center in zone
```

This technique is used throughout the TBMsgWiz application. See: [TBMsgWiz.Exe](#)

History

This feature was implemented in TBMsgBox Version 2.46, November 1994.

Splash Screens

TBMsgBox can be used to create "splash screens". A splash screen is displayed for a second or two then automatically disappears. If you supply a value in the global parameter **MB3D_Splash** then TBMsgBox will simulate the user pressing the first button on the message. The value is specified in milliseconds. For example:

```
MB3D_Splash = 1000
```

causes TBMsgBox to simulate the user pressing the message's first command button after a one second delay.

TBMsgWiz's Help, About [Splash] menu demonstrates this TBMsgBox feature.

Visual Basic Timer - TBMsgBox uses a VB timer. If the **MB3D_Splash** value is greater than zero the routine TimerForSplash is activated after the timer's interval expires.

Help Commands

A help command is added to the TBMsgBox form if variable **MB3D_HelpCommand** is not zero. Four global variables are specified. An example:

```
MB3D_HelpCommand = cdlHelpPartialKey
MB3D_HelpFile = "g:\tbmsgbox\tbmsgbox.hlp"
MB3D_HelpData = "about"      'Search String
MB3D_HelpCaption = "&Learn"  'Alternate Command Caption
```

MB3D_HelpCommand

Any valid WinHelp command can be specified. Most WinHelp commands require that a file name be specified. And some require either a numeric or alphabetic qualifier [data]. In the above example a string ["about"] is specified as the search argument while requesting a Partial Key WinHelp search.

MB3D_HelpFile

The fully qualified file path and name are specified in variable **MB3D_HelpFile**. TBMsgBox uses Window's file search logic to find the file. If the file is not found TBMsgBox ignores the request.

MB3D_HelpData

MB3D_HelpData is defined as a global, variant field. You typically supply a context number when performing context help functions, and a string when performing searches. The data field is not needed with some commands, including the Contents or Index help functions.

MB3D_HelpCaption

You can specify another caption for the help command button. In the above example the word "Learn" is used. Default is "Help". Leave the field "blank" to use the default.

The font specified for other TBMsgBox commands is applied to the Help button. In other words; all TBMsgBox command buttons are made to look the same. See: [Command Captions and Fonts](#).

Visual Basic 4.0 MsgBox Compatibility

Beginning with Visual Basic 4.0 a help file and context number can be included with the MsgBox function. TBMsgBox supports that technique. In addition, TBMsgBox includes a help command with the message, VB does not. See: [Calling TBMsgBox](#) for details.

Mouse "Snap-To" Default Button

TBMsgBox moves the mouse to the default command button. API SetCursorPos is used. Prior to version 3.20 a "dummy" invisible command was moved to the center of the default command. This "work around" is no longer needed.

Technical

Subroutine MB3D_MoveCursorToButton in TBMsgBox.Bas is used to move the mouse pointer. The mouse pointer is positioned slightly right of and below the center of the command to prevent its caption from being obscured.

History

This feature was first implemented in TBMsgBox Version 2.44, November 1994. It was rewritten for Version 3.20, July 1995.

No Buttons

You can display a message without command buttons. For example:

```
MB3D_NoButtons = True
```

The message is terminated when the user presses Esc, Enter or clicks the mouse button.

You could also use a [MB3D_Splash](#) interval to terminate the message after the interval expires. For example:

```
MB3D_Splash = 1000      '1 second
MB3D_NoButtons = True   'No command buttons displayed
TBMsgBox Msg, ...
```

Escape Key

You can still determine a user's response even if the message is displayed without buttons. For example:

```
MB3D_NoButtons = True   'No buttons
Rtn = TBMsgBox(Msg, ..., Ttl)
If Rtn = vbCancel Then  'Esc pressed?
    DoCancelAction      'yes, do cancel action
Else
    DoOtherAction        'no, do other action
End if
```

Technical Information

Buttons Created - The buttons are actually created. That means if you specify the default button as the second button and the second button is "No", vbNo is returned if the user presses "Enter". Pressing Esc always returns vbCancel even if no cancel button was specified.

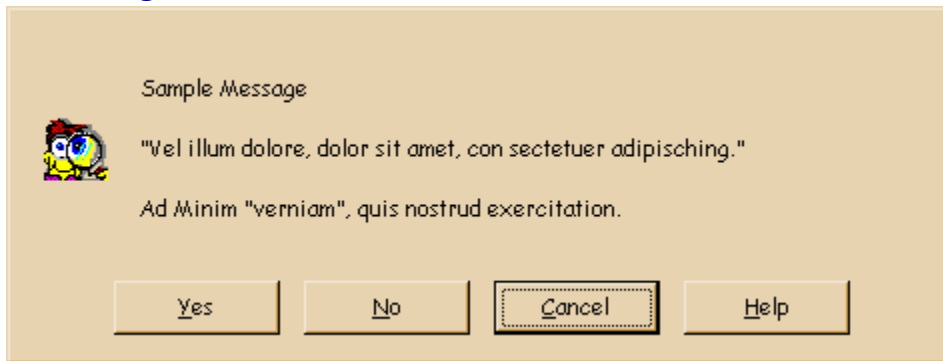
No Title

Messages can be displayed with or without a title bar [caption]. The default is to display the message with a title. Eliminate the title by setting MB3D_NoTitle to true before calling TBMsgBox. For example:

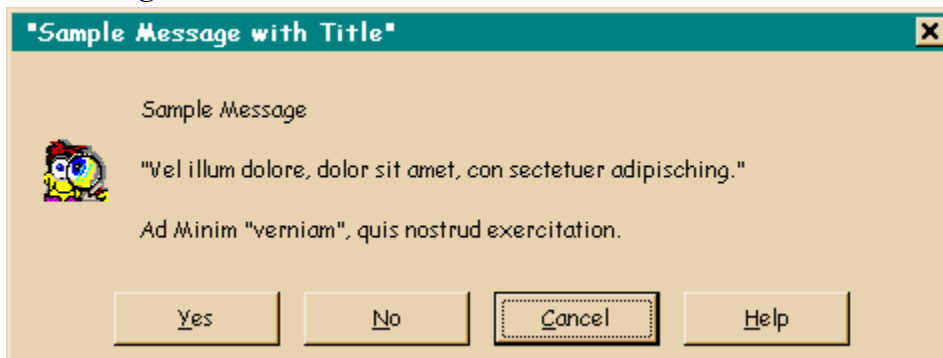
```
MB3D_NoTitle = True
TBMsgBox ....
```

This variable is saved and restored with [MB3D_Globals_Save/Restore](#).

Message without a Title



Message with a Title



Mouse Click

TBMsgBox simulates a click of the invisible default button if the user clicks the mouse in the lower part of a "no button and no title" message.

Technical

TBMsgBox uses Windows API SetWindowLong to change the style of the message box. See subroutine MB3D_SetTitle in module TBMsgBox.Bas for details.

The title is always created. That means you could design messages to contain a title then change the value of MB3D_NoTitle to see how the message looks with or without a title.

History

This feature was implemented in TBMsgBox version 2.62, February 1995. Mouse click in the lower portion of the message was implemented in version 3.10, June 1995. A suggestion from Robert A. Blaney [73157,3543].

VGA Sized Messages

TBMsgBox uses the screen's resolution to determine the message's width. The higher the screen resolution the wider the message. You can override this feature and force all message to be displayed with VGA resolution [640x480]. Set MB3D_VGA_Messages to true before calling TBMsgBox. For example:

```
MB3D_VGA_Messages = True
TBMsgBox ....
```

This variable is saved and restored with [MB3D_Globals_Save/Restore](#).

History

This feature was implemented in TBMsgBox Version 2.52, January 1995.

Hide Rather Than Unload

To dramatically improve performance you can set **MB3D_UnloadAfterDisplay** to False to cause the TBMsgBox form to be hidden after use rather than unloaded. For example:

```
Global Const MB3D_UnloadAfterDisplay = False
```

This constant is defined in module TBMsgBox.Bas. The default setting is "False."

MB3D_Unload

You can force the TBMsgBox form to be unloaded by calling subroutine MB3D_Unload. To demonstrate this feature, a TBMsgWiz.Exe file menu item calls this subroutine. Use it to compare TBMsgBox performance, force the form to be unloaded by using File, Unload then display a TBMsgBox message. You will need to do this after each message is displayed to force the form to be unloaded. MB3D_Unload was added on December 3, 1994 in version 2.48.

Forms.Count Compatibility

Visual Basic will return the number of open forms with the Forms.Count function. If the TBMsgBox form is not unloaded it will be included in the count. If your application uses this function you may need to modify your code to handle this situation. See: [Forms.Count Routines](#) for samples.

Technical Information

Realignment of Code - Beginning with version 2.40 most of the TBMsgBox program logic was moved from the Form [TBMsgBox.Frm] to the Module [TBMsgBox.Bas]. That means the Form does not have to be shown until it has been constructed.

Message Variables - Since the form is not unloaded it means that any variables previously defined in the form, now moved to the module, would not be reset to their default settings. The variables are reset in the module.

Timer Triggers Setting Default Button - A hidden form, when redisplayed [shown], does not always cause an event to occur [Form-Activate is not fired, possibly a VB bug]. This means there is no way to change the default command button. A SetFocus command can't be issued until the form is shown. A timer was added to the form to overcome this limitation. The sequence is:

```
Set the Timer  
Show the Form  
Timer Elapses - Set the Default Button
```

System Modal Window - The only way to gain access to other windows after a system modal window has been loaded is to "destroy" the system modal window. A TBMsgBox system modal message is *always* unloaded - never hidden. System Modal windows are not allowed in 32-bit applications. TBMsgBox does not display message in System Modal mode in 32-bit applications.

MDI Forms.Count Routines

You may find the following routines useful if you hide [rather than unload] TBMsgBox messages and Forms.Count in Multiple Document Interface [MDI] applications.

Count Only Loaded Child Forms

```
Function MDIAllChildCount () As Integer
' Count all the open Child Forms
Dim i As Integer
Dim x As Integer
For i = 0 To Forms.Count - 1
    On Error GoTo MDIAllChildCountError
    If Forms(i).MDIChild Then
        x = x + 1
    End If
MDIAllChildCountSkip:
Next
MDIAllChildCount = x
Exit Function
MDIAllChildCountError:
Resume MDIAllChildCountSkip
End Function
```

Set the Window State for all Loaded Child Forms

```
Sub MDIAllChildSetWindowState (State As Integer)
' Set All child windows to the State Indicated
' For example: State = Minimized
Dim i As Integer
For i = 0 To Forms.Count - 1
    On Error GoTo MDIAllChildSetWindowStateError
    If Forms(i).MDIChild Then
        Forms(i).WindowState = State
    End If
MDIAllChildSetWindowStateSkip:
Next
Exit Sub
MDIAllChildSetWindowStateError:
Resume MDIAllChildSetWindowStateSkip
End Sub
```

Unload all Loaded Child Forms

```
Sub MDIAllChildUnload ()
' Unload all loaded Child Forms
' Need to do in reverse order
Dim i As Integer
For i = Forms.Count - 1 To 0 Step -1
    On Error GoTo MDIAllChildUnloadError
    If Forms(i).MDIChild Then
        Unload Forms(i)
    End If
MDIAllChildUnloadSkip:
Next
Exit Sub
MDIAllChildUnloadError:
Resume MDIAllChildUnloadSkip
End Sub
```

Converting from VB's MsgBox to TMsgBox

Conversion

Convert VB MsgBox statements to [TMsgBox](#) subroutines. Convert VB MsgBox functions to [TMsgBox](#) functions. For example the function:

```
iRtn = MsgBox(TheMessage, vb..., TheTitle)
```

when converted becomes:

```
iRtn = TMsgBox(TheMessage, vb..., TheTitle)
```

Replace MsgBox with TMsgBox throughout you application. **Warning**, you should make a backup of your application before applying any global change.

Add Files

Add, if necessary, the following files to your VB project:

TMsgBox.Frm	Message Box Form
TMsgBox.Bas	Subroutines, Functions and Globals

Customize the Message

You may wish to "customize" the message by changing message alignment, command captions, font properties, message icons or adding a help button. See: [Message Alignment and Fonts](#), [Command Captions and Fonts](#), [Alternate Icons](#) or [Help Commands](#) for details.



The MsgBox Wizard

Click one of the following topics:

[TBMsgWiz.Exe - The MsgBox Wizard](#)

[Message Alignment And Fonts](#)

[Command Captions and Fonts](#)

[Selecting a Font](#)

[Selecting An Alternate Icon](#)

[Selecting a Help File](#)

[Generating TBMsgBox VB Source Code](#)

[TBMsgWiz Settings are Saved](#)

[Displaying What's New](#)



TBMsgWiz.Exe - The MsgBox Wizard

TBMsgWiz is used to generate Visual Basic source code that you can copy into your project. The source code is written to the Clipboard. TBMsgWiz also demonstrates the similarities and differences between VB's MsgBox and TBMsgBox. You may wish to review VB's help topic MsgBox Function. TBMsgBox replicates the functionality of VB's MsgBox. TBMsgWiz is written in VB 4.0, 32-bit. It can be accessed and executed in the VB 4.0 16-bit environment. Click on the following form for additional information about each section of this screen. There are 16 [hot spots](#) on this screen.

The Bridge's MsgBox Wizard

File Sample Message What's New.. Help Test

Buttons <ul style="list-style-type: none"><input type="checkbox"/> MB3D_NoButtons<input type="radio"/> vbOKOnly<input type="radio"/> vbOKCancel<input type="radio"/> vbAbortRetryIgnore<input checked="" type="radio"/> vbYesNoCancel<input type="radio"/> vbYesNo<input type="radio"/> vbRetryCancel<input type="checkbox"/> Include a Help Button	Icons <ul style="list-style-type: none"><input type="radio"/> vbNone<input type="radio"/> vbCritical<input type="radio"/> vbQuestion<input type="radio"/> vbExclamation<input checked="" type="radio"/> vbInformation<input type="checkbox"/> Use an Alternate Icon
Default Button <ul style="list-style-type: none"><input type="radio"/> vbDefaultButton1<input type="radio"/> vbDefaultButton2<input checked="" type="radio"/> vbDefaultButton3	Modal Type <ul style="list-style-type: none"><input checked="" type="radio"/> vbApplicationModal<input type="radio"/> vbSystemModal
MsgBox or x=MsgBox() <ul style="list-style-type: none"><input type="radio"/> Subroutine<input checked="" type="radio"/> Function	

Message: Sample Message [Edit..]

☒ Title: "Sample Title" [vbCancel [2]]

[VB MsgBox] [TB MsgBox] [Fonts...] [Captions...]

Select desired option buttons then click the VB MsgBox or TBMsgBox commands to display the standard VB or TBMsgBox message. The generated source code is written to the Clipboard each time you click one of these commands.

Enter new message or title text to display a message containing that text.

Use the Fonts command to change [message alignment or font properties](#). Use the Captions command to change [command captions and font properties](#).



Message Alignment And Fonts

You can change TBMsgBox message's alignment and font properties with this form. Click on the following form for additional information about each section. There are 12 [hot spots](#) on this screen.

The dialog box has a title bar with the text 'Select a Message Font and Alignment' and standard window controls. Below the title bar is a menu bar with 'File', 'Alignment', 'Font', and 'Help'. The main area displays the current font settings: 'Comic Sans MS, 9, Regular'. Below this, there is a text box containing instructions: 'Click the Left, Center or Right option to change the message text alignment.' and 'Click the 'Select a Different Font...' command to change the message's font properties.' Below the text box are three radio buttons for alignment: 'Left', 'Center' (which is selected), and 'Right'. Below the radio buttons is a button labeled 'Select a Different Message Font...'. At the bottom of the dialog are four buttons: 'Default', 'OK', 'Cancel', and 'Help'.

Note: While operating in 32-bit mode, TBMsgBox always uses the message font defined in Windows 95 non-client metrics. That means that no matter what font is chosen the message font defined in non-client metrics will be used. Non-client metric information is not accessed if TBMsgBox is operating in 16-bit mode.

This screen is centered within a zone defined using [MB3D Zone](#).



Command Captions and Fonts

You can change TBMsgBox command's captions and font properties with this form. Click on the following form for additional information about each section. There are 11 [hot spots](#) on this screen.

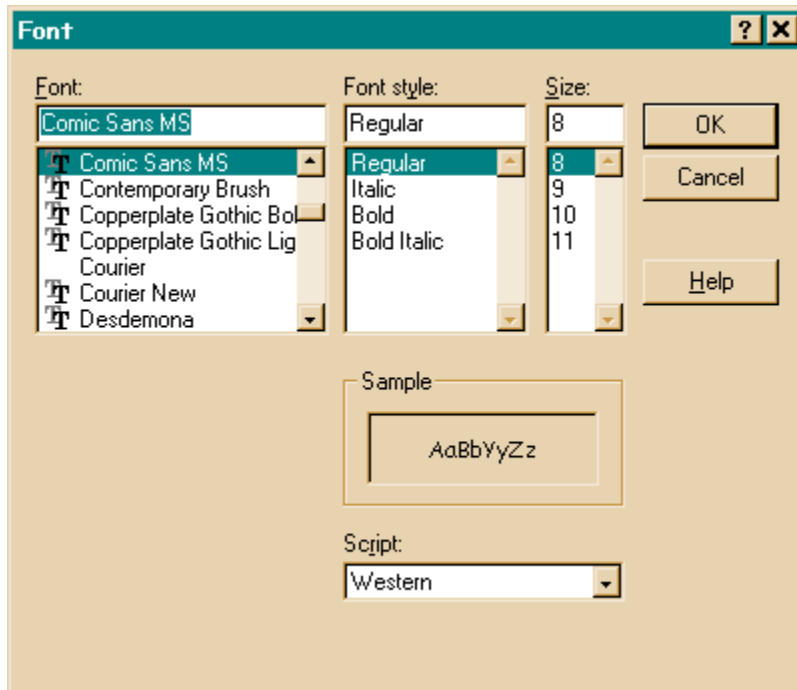
Note: While operating in 32-bit mode, TBMsgBox always uses the menu font defined in Windows 95 non-client metrics. That means that no matter what font is chosen the menu font defined in non-client metrics will be used. Non-client metric information is not accessed if TBMsgBox is operating in 16-bit mode.

This screen is centered within a zone defined using [MB3D Zone](#).



Selecting a Font

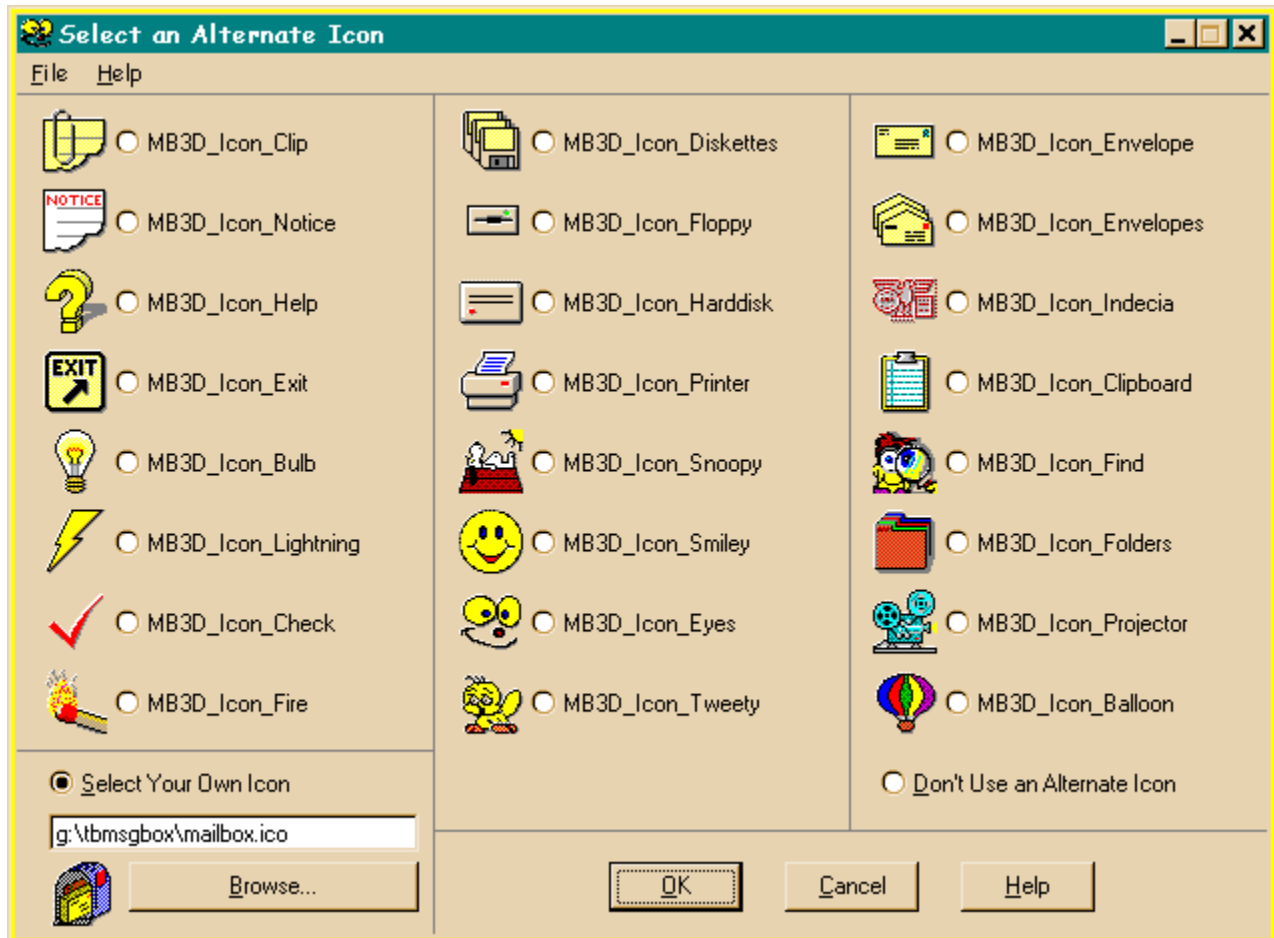
The following Windows 95 Common Dialog screen is displayed when you click the [Select a Different Font](#) command:





Selecting an Alternate Icon

The following screen is displayed when you click the "Use an Alternate Icon" option or menu item File, Select an Alternate Icon. Click on the following form for additional information about each section. There are 6 [hot spots](#) on this screen.



This screen is centered within a zone defined using [MB3D_Zone](#).



Selecting a Help File

The following screen is displayed when you click the "Include a Help File" option or menu item File, Select a Help File. Click on the following form for additional information about each section. There are 11 [hot spots](#) on this screen.

The screenshot shows a Windows-style dialog box titled "Setup the MsgBox Help Button". It has a menu bar with "File" and "Help". The dialog contains several input fields and buttons:

- "WinHelp Command:" with a dropdown menu showing "Help_PartialKey [Search uses Data string]" and a button "&H105".
- "Help File Name:" with a text field containing "G:\TBMSGBOX\Tbmsgbox.hlp" and a "Browse..." button.
- "WinHelp Data:" with a text field containing "about".
- "Command Caption:" with a text field containing "&Learn" and a button "Learn".
- At the bottom, there are four buttons: "Test", "OK" (which is highlighted with a dashed border), "Cancel", and "Help".

This screen is centered within a zone defined using [MB3D_Zone](#).



Generating TBMsgBox Source Code

If File menu item [Write Generated Code to the Clipboard](#) is checked, TBMsgWiz will generate the Visual Basic code used to create the sample message. The code is written to the clipboard. You can paste the code into your VB application.

Sample Generated Code

TBMsgWiz generated the following code. VB comments have been eliminated to make the code easier to read.

```

' 1996/04/11 19:46:07 , Code generated by TBMsgBox 4.00...
Dim MB3D_Msg as String
Dim MB3D_Title as String
Dim MB3D_Type as Integer
Dim MB3D_ReturnCode as Integer
Dim MB3D_SaveArray() as Variant
MB3D\_Globals\_Save MB3D_SaveArray()
MB3D_Msg = ""
MB3D_Msg = MB3D_Msg & "Sample Message" & vbCr
MB3D_Msg = MB3D_Msg & vbCr
MB3D_Msg = MB3D_Msg & Chr$(34)
MB3D_Msg = MB3D_Msg & "Vel illum dolore, sectetuer adipiscing."
MB3D_Msg = MB3D_Msg & Chr$(34)
MB3D_Msg = MB3D_Msg & vbCr
MB3D_Msg = MB3D_Msg & vbCr
MB3D_Msg = MB3D_Msg & "Ad Minim "
MB3D_Msg = MB3D_Msg & Chr$(34)
MB3D_Msg = MB3D_Msg & "verniam"
MB3D_Msg = MB3D_Msg & Chr$(34)
MB3D_Msg = MB3D_Msg & ", quis nostrud exercitation."
MB3D_Title = ""
MB3D_Title = MB3D_Title & Chr$(34)
MB3D_Title = MB3D_Title & "Sample Title"
MB3D_Title = MB3D_Title & Chr$(34)
MB3D_Type = vbYesNoCancel
MB3D_Type = MB3D_Type + vbInformation
MB3D_Type = MB3D_Type + vbDefaultButton3
MB3D_Alignment = 2
MB3D_FontName = "Comic Sans MS"
MB3D_FontSize = 9
MB3D_FontNotBold = True
MB3D_Commands_FontName = "Comic Sans MS"
MB3D_Commands_FontSize = 9
MB3D_Commands_FontNotBold = True
MB3D_Icon_Which = 15
MB3D_HelpCommand = cdlHelpHelpOnHelp
'MB3D_Zone Me
'MB3D_VGA_Messages = False
'MB3D_NoTitle = False
'MB3D_BevelWidth = 1
MB3D_ReturnCode = TBMsgBox (MB3D_Msg, MB3D_Type, MB3D_Title)
MB3D\_Globals\_Restore MB3D_SaveArray()
' Add supporting instructions to these generated statements:
If MB3D_ReturnCode = vbYes Then
End If
If MB3D_ReturnCode = vbNo Then
End If
If MB3D_ReturnCode = vbCancel Then
End If

```

Beginning with version 2.81, April 1995, output is created for VB MsgBox as well as TBMsgBox.



TBMsgWiz Settings are Saved

When you exit TBMsgWiz options and settings you selected are saved in TBMsgWiz.Ini. These settings are restored the next time the program is executed.

A sample Ini file:

```
; TBMsgWiz.Ini - TBMsgWiz Initialization File
; 96/04/02 Created this date.

[Window]
Left=4222
Top=2018
Screen Width=15360
Screen Height=11520

[Preferences]
Output to Clipboard=True
Display What's New Message=2.80 True
Save Session Options=True

[Saved Command Button Names]
MB3D_UD_OK=&OK
MB3D_UD_Cancel=&Abbrechen
MB3D_UD_Abort=&Abbrechen
MB3D_UD_Retry=&Wiederholen
MB3D_UD_Ignore=&Ignorieren
MB3D_UD_Yes=&Ja
MB3D_UD_No=&Nein

[Saved Message Font Information]
MB3D_Alignment=Left_Justify
MB3D_FontName=
MB3D_FontSize=0
MB3D_FontBold=False
MB3D_FontItalic=False

[Saved Command Button Font Information]
MB3D_Commands_FontName=
MB3D_Commands_FontSize=0
MB3D_Commands_FontBold=False
MB3D_Commands_FontItalic=False

[Saved Help Information]
MB3D_HelpCommand=cdlHelpPartialKey
MB3D_HelpFile=g:\TBMsgBox\TBMsgBox.hlp
MB3D_HelpData=about
MB3D_HelpCaption=&Learn

[Saved Message and Title]
MB3D_Message=+++Sample Message||\|\|\|"Vel illum...
MB3D_Title=+++ "Sample Title"+++

[Saved Zone Information]
MB3D_ZoneLeft=4222
MB3D_ZoneTop=2018
MB3D_ZoneWidth=6915
MB3D_ZoneHeight=7035

[Saved Miscellaneous Information]
MB3D_Splash=0
MB3D_NoButtons=False
MB3D_Icon_Which=MB3D_Icon_Smiley
MB3D_Icon_FileName=
MB3D_VGA_Messages=False
MB3D_NoTitle=False

[Saved TBMsgWiz Settings]
MG3D_Button=MB_YesNoCancel
MG3D_Default=MB_DefButton1
```

```
MG3D_Icon=MB_IconInformation  
MG3D_Modal=MB_ApplModal  
MG3D_SubFun=Function
```

Technical

The Ini file is saved in TBMMsgWiz's application directory unless it is read-only, in which case the file is stored in the Windows directory. A new Ini file is created if it does not already exist. You could delete this file to revert to factory defaults.

History

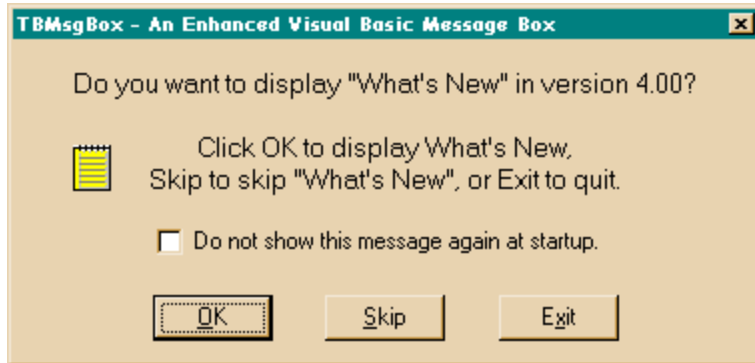
Saving the TBMMsgWiz location was implemented in TBMMsgBox Version 2.47, November 1994. Saving selected settings was implemented in version 2.74, March 1995. Saving the Ini file in the Windows directory was implemented in version 4.00, March 1996.



Displaying What's New

You can display the historical record of TBMsgBox development. Click menu item **"What's New"**.

The following dialog is displayed the first time you run TBMsgWiz:



Click the check box to prevent this message from being displayed again.

Registration and Ordering TBMsgBox

Click one of the following topics:

[Registration and Ordering TBMsgBox](#)

[Registered Version License](#)

[Upgrade Policy](#)

[About TBMsgBox](#)

[Other Products](#)

[Update Information](#)

Registration and Ordering TMsgBox

You can order TMsgBox's Visual Basic source code:

Using CompuServe's Shareware Registration [GO SWREG, 3805], or

Using the Public (software) Library

Item number: 14049,

Orders: 800-242-4775

Information: 713-524-6394, or

Directly from The Bridge:

Send a CompuServe message to 71662,205, or

Send an Internet message to 71662.205@CompuServe.com, or

Mail this form to the address shown below, or

Fax this form to 415-737-0871, or

Phone 415-737-0870

Be sure to include credit card information.

Click the Print button to print this order form.

TMsgBox Components

Click here to view [TMsgBox Source Components](#).

Ship TMsgBox To

Name: _____
Company: _____
Address: _____
City/State/Code _____
Country _____
Phone: _____
CompuServe: _____

Price:

Product: \$19.95, California purchasers add appropriate sales tax.

Price is subject to change without notice.

Shipping: \$3.95

Upgrade: Click here to review the [Upgrade Policy](#).

Ship Via

CompuServe _____ US Mail _____

Shipped via CompuServe unless you specify US Mail.

Registered User License

Click here to review the [Registered User's License](#).

Payment Method

Check _____ Visa _____ MasterCard _____

If Credit Card, Number: _____,

Expires: _____

If Check, make payable to and Mail to:



Inc

913 Hacienda Way
Millbrae, CA 94030
415-737-0870
Fax: 415-737-0871
CompuServe: 71662,205

Last updated: February 26, 1996

Version: 4.01

Version Date: March 1996

TBMsgBox - Registered User License

You should carefully read the following terms and conditions before using TBMsgBox. Its use indicates your acceptance of these terms and conditions. If you do not agree with them do not use the software.

Licensed

The Bridge, Inc., (hereafter The Bridge), hereby grants a License to the Licensee to use the TBMsgBox (hereafter The Product):

1. on a single computer system that contains only one central processing unit that can access The Product;
2. to create application software using The Product, including the trade secrets and confidential information contained therein, and to incorporate in such application software the Visual Basic source code provided with The Product.
3. to make two copies of The Product for archival purposes only.

Not Licensed

No License is granted to:

1. transfer to any third-party The Product's Visual Basic source code.
2. to create derivative works of The Product.

Termination

This license is effective until terminated. Licensee may terminate this License by destroying all copies of The Product in his/her possession. The Bridge may terminate this License immediately if Licensee fails to observe any of the terms of this license. If terminated, Licensee must destroy all copies of The Product immediately, or return them to The Bridge.

Copyright Ownership

The Product is owned by The Bridge and is protected under the copyright laws of the United States and international treaty provisions. Notwithstanding the copyright, The Product contains trade secrets and confidential information of The Bridge which remains the property of The Bridge. The Bridge reserves all rights not expressly granted to the Licensee in this License.

Limited Warranty

The Bridge warrants that The Product will function substantially as described in The Product's Help file documentation [TBMsgBox.Hlp] if used in accordance with instructions contained therein.

EXCEPT FOR THE LIMITED WARRANTY EXPRESSLY PROVIDED IN THIS SOFTWARE LICENSE AGREEMENT AND LIMITED WARRANTY, NO OTHER WARRANTIES ARE PROVIDED BY THE BRIDGE. THE BRIDGE SPECIFICALLY EXCLUDES ANY AND ALL IMPLIED WARRANTIES, INCLUDING WARRANTIES OF COMMERCIAL VIABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND LIMITS YOUR REMEDY TO RETURN OF THE FAULTY SOFTWARE WITHIN 90 DAYS OF RECEIPT. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES, SO THE ABOVE LIMITATION MAY NOT APPLY TO LICENSEE.

THE BRIDGE DISCLAIMS ANY LIABILITY FOR INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION

MAY NOT APPLY TO LICENSEE.

Interpretation

This License shall be construed in accordance with the laws of the State of California and constitutes the sole and exclusive agreement and understanding between the Licensee and The Bridge.



Inc.

913 Hacienda Way
Millbrae, CA 94030

CompuServe: 71662,205
415-737-0870
Fax: 415-737-0871
Copyright 1996

Upgrade Policy

Registered Users Prior to 4.00

TBMsgBox is available for \$9.95 and a \$3.95 shipping charge to registered users of earlier versions. Please send a message to The Bridge [CompuServe: 71662,205] noting your VISA or MasterCard number, with expiration date. Or send a check to:



Inc.

913 Hacienda Way
Millbrae, CA 94030

Or fax the [Order Form](#) to 415-737-0871.

About TBMsgBox

Click the right mouse button on a TBMsgBox Icon, or press F12, while displaying a TBMsgBox message to display this "About" box:



The message can also be displayed from the TBMsgWiz's help menu [Help, About].

Other Products

Other products available from The Bridge in the CompuServe MSBasic forum include:

HolCal - A Holiday and Event Calendar



A full function holiday and event calendar. Displays holidays and events using tool tips. Use it as a stand alone calendar and while minimized as a clock. Or use it to communicate with your application via DDE. Source code is available for \$19.95. MSBasic forum file name: HolCal.Zip. Version 3.0 released September 1995.

DemoCt3D

Demonstrates the API's needed to call Ctl3D.DLL. Freeware. File name: DmoC3D.Zip.

KeyState

According to information found in the Microsoft Developer's Network CD 10, KeyStat.vbx can cause a GPF under certain conditions. We developed this freeware workaround which you may wish to use in place of the KeyStat.vbx. File name: KeySta.Zip.

Shade

A demo program that shows how to shade a form - similar to the Window's Setup screen. Freeware. File name: Shade.Zip.

SetTab

A demo program that shows how to set tabs in a text box. Freeware. File name: SetTab.Zip.

TstLay

Two functions that will save and restore a TrueGrid layout in a binary file. Freeware. File name: TstLay.Zip.

UppLow

A demo program that shows how to use the Windows API to force text box input to all upper or lower case. Freeware. File name: UppLow.Zip.

Update Information

Last updated: February 26, 1996

File: g:\tbmsgbox\help\tbmsgbox.doc

Version: 4.01

Version Date: March 1996

Hot Spots

Most of the forms displayed in this help file contain "hot spot" areas. The mouse cursor changes to a "hand" as it moves over a "hot spot". Click the mouse on the "hot spot" to pop-up information associated with that area of the form.

Technical Note - You can use the tab key to locate "hot spots". Then, press the Enter key to display information associated with the highlighted area.

Calling Subroutine TBMsgBoxSub

A typical TBMsgBoxSub calling sequence:

```
TBMsgBoxSub message, type, title
```

For example:

```
TBMsgBoxSub "A Message", MB_IconStop, "Title"
```

will create a three dimensional message box with a message and title as well as display a stop icon and an OK button.

Calling Function TBMsgBoxFun

A typical TBMsgBoxFun calling sequence:

```
RtnCode = TBMsgBoxFun (message, type, title)
```

For example:

```
RtnCode = TBMsgBoxFun ("Message", MB_OKCancel, "Title")
```

will create a three dimensional message box with a message and title as well as an OK and a Cancel button. If the user clicks the OK button RtnCode will contain IDOK [1], otherwise it will contain IDCANCEL [2].

VB40032.DLL and VB40016.DLL - VB Dynamic Link Libraries

TBMsgWiz is written in 32-bit Visual Basic. It also supports 16-bit mode. The appropriate run-time DLL is needed to run a compiled version of the program. Make sure the Windows/System directory contains a copy of these Dynamic Link Libraries.

ThreeD16/32.OCX - Three Dimensional Controls

ThreeD16.OCX and ThreeD32.OCX are distributed with Visual Basic. TBMsgWiz uses this OCX. TBMsgBox does **not** use.

ComDlg16/32.OCX - Common Dialog Control

The Common Dialog control provides support for a number of commonly used functions. TBMsgWiz uses CMDialog.VBX to select fonts and locate an alternate icon. It is **not** used by TBMsgBox.

Buttons

Select the command button configuration to be displayed with the message box. In this example the Yes, No and Cancel buttons will be displayed.

MB3D_NoButtons

Select "MB3D_NoButtons" to display the message without buttons. See: [No Buttons](#) for details.

Include a Help Button

Select "Include a Help Button" to initiate the Help Button Selection process. See: [Selecting a Help File](#) for details.

Default Button

If more than one command button is displayed, set the default button by selecting one of the Default Button options. In the example the third [Cancel] button becomes the default.

Icons

Select an Icon to be displayed with the message. Click the **Use an Alternate Icon** check box to select an alternate icon. See: [Selecting an Alternate Icon](#) for details.

Modal Type

The Windows 16-bit API supports either Application wide or System wide modal window display. An Application wide modal form must be "unloaded" before other forms in that application can be selected, however; *other applications* can be selected. A System wide modal window must be "unloaded" before any other application can be selected. System wide modal operation is not valid with 32-bit mode operating systems.

In 16-bit operation, TBMsgBox uses Window's API SetSysModalWindow to set a system wide modal message. There are a few differences between the VB MsgBox and TBMsgBox message displayed in system modal form. Differences:

VB MsgBox

- No Title or Control Box in some cases.

- No border in some cases.

TBMsgBox

- If MB3D_NoTitle=False, always has a Title and Control Box.

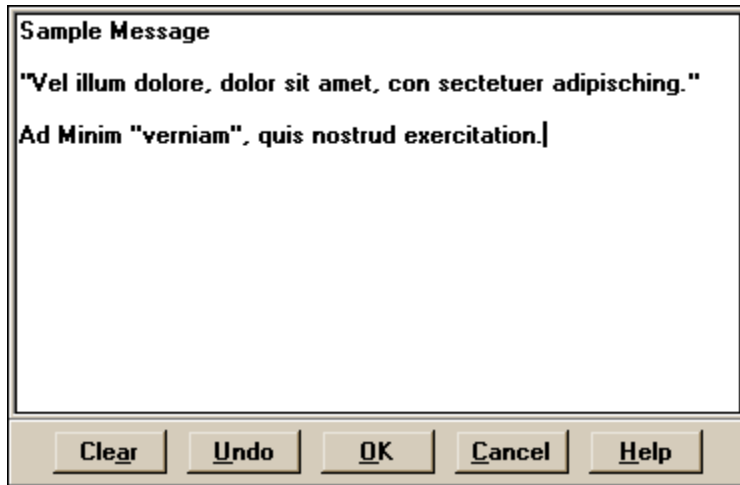
- Always has a border.

Subroutine or Function

Select either a subroutine or function option. In this example the Function option was selected.

Message

Use the Edit button to expand the message box. You can enter a very large message string. Click Clear to clear the message, Undo to restore the message, OK to accept the changed message, Cancel to revert to the prior message text, or Help to display this helpful information. A sample form:



Sample Message

"Vel illum dolore, dolor sit amet, con sectetuer adipiscing."

Ad Minim "verniam", quis nostrud exercitation.]

Clear **Undo** **OK** **Cancel** **Help**

[History](#) - Undo was added in version 2.87, May 1995.

Title

If checked, enter any desired title. Maximum length is 78 characters. If blank, TBMsgBox will first use App.Title then App.EXENAME. Display a "blank" title by entering at least one space.

If unchecked, no title is displayed. And the line:

```
MB3D_NoTitle = True
```

is added to generated Visual Basic code.

See [No Titles](#) for details.

Return Code

Return Code - The value returned by the TBMsgBoxFun indicates which button was selected. See: [TBMsgBox](#) for valid return codes.

Standard VB Message

Click the VB Message command to display a standard Visual Basic message.

TB Message

Click the TBMessage command to display the message created by it.

Fonts

Click this key to change message alignment and font properties. See: [Alignment and Font Selection](#) for details.

Captions

Click this key to change command captions and font properties. See: [Command Captions and Fonts](#) for details.

File

<u>M</u> essage Fonts...	Ctrl+O
<u>C</u> ommand Captions...	Ctrl+C
<u>S</u> elect an Alternate Icon...	Ctrl+S
Select a <u>H</u> elp File...	Ctrl+H
✓ <u>W</u> rite Generated Code to the Clipboard	Ctrl+W
✓ <u>S</u> ave Selected Options in <u>I</u> ni File	Ctrl+I
Reset all Options to <u>F</u> actory Settings	Ctrl+R
Start Clipboard <u>V</u> iewer	Ctrl+V
<u>C</u> enter Me	F9
<u>U</u> nload the MsgBox Form	
Maximum Size is VGA Messages [640x480]	
<u>E</u> xit	Alt+F4

[Message Fonts](#) - [Command Captions](#) - [Alternate Icons](#) - [Help File](#)

[Write Generated Code to the Clipboard](#)

[Save Selected Options in Ini File](#) - [Reset all Options to Factory Settings](#)

[Start Clipboard Viewer](#) - [Center Me](#)

[Unload the TMsgBox Form](#)

[Maximum Size is VGA Message](#)

Font, Caption, and Icon Selection

Use the first section of the file menu to start Font [and alignment] message selection; Caption [font and text] selection, Alternate Icon selection and Help button selection.

Write Generated Code to the Clipboard

Use the second part of the file menu to start or end creation of VB source code. You can also start the Clipboard viewer. The Clipboard Viewer is started as an icon [minimized]. Select this option again to set the Clipboard Viewer mode to "normal". "Manually" close the Clipboard Viewer after you are finished viewing the generated output.

Your output generation preference is stored in the TBMsgWiz.Ini file, for example:

```
[Preferences]
Output to Clipboard=True
```

Save Selected Options in Ini File

If checked, settings you selected while using TBMsgWiz are saved in the TBMsgWiz.Ini file.

Reset all Options to Factory Settings

Reset all options to their factory settings. Use this menu item to restore all options to the settings originally provided with TBMsgWiz. You could accomplish the same result by deleting the TBMsgWiz.Ini file.

Center Me

Center this TBMsgWiz form.

Unload the TBMsgBox Form

The TBMsgBox form is usually not unloaded after being displayed, it is hidden. Select this menu item to force an unload of the TBMsgBox form.

This feature is primarily intended to demonstrate the difference in display speed. For example: Display a message a few times, then select this menu item and display the message again. Notice the difference in the display speed.

Maximum Size is VGA Message

If checked, TBMsgWiz displays messages using VGA resolution, 640 by 480. And the line:

```
MB3D_VGA_Messages = True
```

is added to generated Visual Basic code. See [VGA Sized Messages](#) for details.

Display Sample Message

Use these menu items in lieu of the VB Message or TBMsgBox commands.

What's New

Select this menu item to display the "What's New" messages.

Font Name and Properties

The selected font name, size, type [regular, bold, italic, bold italic].

Sample Message

The message displayed by TBMsgBox will look like the text displayed here. The text style changes as you select different font properties.

Message Alignment

Click a Left, Center or Right option button to change the message's alignment. The menu item, Alignment, can be used to accomplish the same results.

A centered message is shown in this example.

Select a Different Font

Click this command button to display a font selection dialog screen. See: [Selecting a Font](#) for details.

Use Default Message Settings

Click this button to reset the message font and alignment properties to their "factory" settings.

OK - Accept the New Settings

Click this button to use the "new" settings. A sample message, using the new settings, is displayed.

Cancel Changes Made to the Settings

Click this button to cancel any changes. Return to the main screen.

Help

Display helpful information.

File Menu Item

File menu items are OK, Cancel and Exit. Selecting OK or Cancel causes the same action as the corresponding command button. Select "Exit" to immediately terminate TBMsgWiz.

Default Captions

These are the default command captions. They are used by TBMsgBox unless you specify a replacement.

Replacement Captions

You can enter any replacement caption in the text box provided. The text is moved into the command caption as you type it.

Factory <u>D</u> efault	Ctrl+D
Some <u>S</u> amples	Ctrl+M
<u>S</u> panish [Maybe]	Ctrl+S
<u>F</u> rench	Ctrl+F
<u>G</u> erman	Ctrl+G
<u>D</u> utch	Ctrl+U
<u>P</u> ortugese	Ctrl+P
✓ <u>I</u> talian	Ctrl+I

Sample captions are supplied with TBMsgWiz. Click the Captions menu item then select one of the supplied sets.

Use Default Settings

Click this button to reset the command captions and font properties to their "factory" settings.

Captions

Factory <u>D</u> efault	Ctrl+D
Some <u>S</u> amples	Ctrl+M
<u>S</u> panish [Maybe]	Ctrl+S
<u>F</u> rench	Ctrl+F
<u>G</u> erman	Ctrl+G
<u>D</u> utch	Ctrl+U
<u>P</u> ortugese	Ctrl+P
✓ <u>I</u> talian	Ctrl+I

Sample command captions are supplied with TBMsgWiz. Select one of the samples from this menu. Or select "Factory Default" to reset the commands to their "factory" setting.

French - Thanks to Roger Doyon, CIS 74361,3542, who reviewed the French captions. The "updated" captions are included with TBMsgBox version 2.44 and later, first distributed on November 6, 1994.

German - Thanks to Holger Thornagel, CIS 100407,356, who supplied the German captions. They are provided with TBMsgBox version 2.72 and later, first distributed on March 12, 1995.

Spanish - The Spanish captions were selected, by the author, from a language translation dictionary. They should be reviewed by someone proficient in the language. The author would appreciate any help you could provide. He would like to include other language captions with TBMsgBox.

Dutch - Thanks to Bert Pol, Makandra Software Productions, 2134 JE Hoffddorp NH, Holland, for the Dutch captions. [95/11/04]

Portuguese [Brazilian] - Thanks to Jose Garcia Junior, 102105,2421 for the Portuguese - Brazilian captions. [96/01/04]

Italian - Thanks to Carlo Queirolo, INTERNET:queirolo@mbox.vol.it for the Italian captions. [96/01/11]

Font

Select this menu item to display the font selection dialog.

Font Name

Select a font name from the list of fonts.

Font **Bold** and *Italic*

Select a regular, italic, bold, or bold italic font style from the list of styles. Not all four styles apply to all fonts. In this example 'Bold' is highlighted.

Font Size

Select a font size. Although the font size is restricted in TBMsgWiz, TBMsgBox will display any reasonable size. In this example '8' is highlighted.

OK and Cancel Commands

Click the **OK** command to accept the changed font properties.

Click **Cancel** to ignore any changes.

Font Sample

A sample of the font is displayed in this box as you select different font properties.

Command Captions

```
Global MB3D_UD_OK As String
Global MB3D_UD_Cancel As String
Global MB3D_UD_Abort As String
Global MB3D_UD_Retry As String
Global MB3D_UD_Ignore As String
Global MB3D_UD_Yes As String
Global MB3D_UD_No As String
```

Message Alignment and Fonts

```
Global MB3D_Alignment As Integer
Global MB3D_FontName As String
Global MB3D_FontSize As Single
Global MB3D_FontBold As Integer
Global MB3D_FontItalic As Integer
```

Command Fonts

```
Global MB3D_Commands_FontName As String
Global MB3D_Commands_FontSize As Single
Global MB3D_Commands_FontBold As Integer
Global MB3D_Commands_FontItalic As Integer
```

Splash Time, in milliseconds

```
Global MB3D_Splash As Integer
```

No Buttons

```
Global MB3D_NoButtons As Integer
```

No Title

```
Global MB3D_NoTitle as Integer
```

Alternate Icons

```
Global MB3D_Icon_Which As Integer
Global MB3D_Icon_FileName as String
```

Help Command Variables

```
Global MB3D_HelpCommand As Integer
Global MB3D_HelpFile As String
Global MB3D_HelpData As Variant
Global MB3D_HelpCaption As String
```

Zone Centering Variables

```
Global MB3D_ZoneLeft As Long
Global MB3D_ZoneTop As Long
Global MB3D_ZoneWidth As Long
Global MB3D_ZoneHeight As Long
```

VGA Sized Messages

```
Global MB3D_VGA_Messages as Integer
```

Alternate Icon Constants

```
Global Const MB3D_Icon_Clip = 1
Global Const MB3D_Icon_Notice = 2
Global Const MB3D_Icon_Help = 3
Global Const MB3D_Icon_Exit = 4
Global Const MB3D_Icon_Bulb = 5
Global Const MB3D_Icon_Lightning = 6
Global Const MB3D_Icon_Check = 7
Global Const MB3D_Icon_Fire = 8
Global Const MB3D_Icon_Diskettes = 9
Global Const MB3D_Icon_Floppy = 10
Global Const MB3D_Icon_Harddisk = 11
Global Const MB3D_Icon_Printer = 12
Global Const MB3D_Icon_Snoopy = 13
Global Const MB3D_Icon_Smiley = 14
Global Const MB3D_Icon_Eyes = 15
Global Const MB3D_Icon_Tweety = 16
Global Const MB3D_Icon_Envelope = 17
Global Const MB3D_Icon_Envelopes = 18
Global Const MB3D_Icon_Indecia = 19
Global Const MB3D_Icon_Clipboard = 20
Global Const MB3D_Icon_Find = 21
Global Const MB3D_Icon_Folders = 22
Global Const MB3D_Icon_Projector = 23
Global Const MB3D_Icon_Balloon = 24
```

Alternate Icons

Select one of the alternate icons then click OK to display a sample message using the selected icon.

Select Your Own Icon

Use the Browse command to locate an icon, or type the fully qualified file name in the text box. Click the OK command to display a sample message using the selected icon.

Don't Use an Alternate Icon

Click this option to discontinue alternate icon processing. Click OK to return to the main screen.

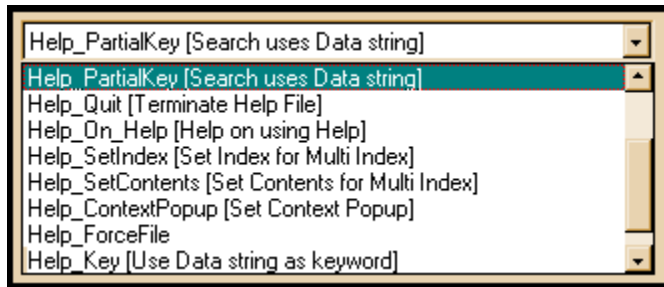
Global Variables Used With TBMsgBox Help

Help Command Variables

Global MB3D_HelpCommand As Integer	'command
Global MB3D_HelpFile As String	'file, fully qualified
Global MB3D_HelpData As Variant	'data
Global MB3D_HelpCaption As String	'command caption

WinHelp Command

Select a WinHelp command from this combo box. Other controls are disabled until a command is selected.
Some of the entries in the combo box:



WinHelp Command Value

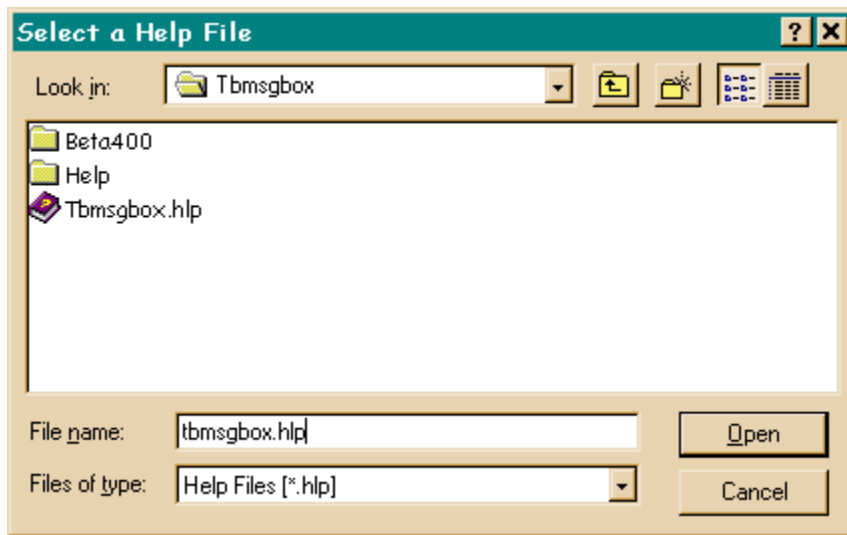
The numeric value [in hexadecimal notation] of the selected WinHelp command.

Help File

The fully qualified file path and name of the help file to be opened by WinHelp.

Browse

Click the browse command to search for a file. A sample:



WinHelp Data

Enter a search string for Partial Key searches or a context number for context sensitive help. Not required for other WinHelp commands.

Help Command Caption

Enter an alternate help command caption. Default is "Help".

Help Sample Button

This is a sample help command button. If an alternate font has been specified then this command is displayed with that command attribute. See: [Command Captions and Fonts](#) for details.

Click this command to "test" the help function. The same as clicking the "Test" button.

Test

Call WinHelp using the variables specified via this form.

OK

Accept the items specified via this form. Build a sample message which contains the help button.

Cancel

Terminate this function and ignore changes made via this form.

Help

Display this help topic.

TBMsgBox Source Components

Used in Your Application:

TBMsgBox.Frm	Message Form
TBMsgBox.Frx	Binary Information, Mostly Icons
TBMsgBox.Bas	Subroutines, Functions, Globals

Used with TBMsgWiz:

TBMsgBox.Ico	Icon
TBMsgBox.Hlp	This Help File
TBMsgBox.Ghc	Help File Constants
TBMsgDef.Ghc	
TBMsgWiz.Frm	Generate VB Code and Demo Program
TBMsgWiz.Frx	
TBMsgHlp.Frm	Help Command Processor
TBMsgHlp.Frx	
TBMsgFnt.Frm	Message Fonts
TBMsgFnt.Frx	
TBMsgCap.Frm	Command Captions
TBMsgCap.Frx	
TBMsgIco.Frm	Alternate Icons
TBMsgIco.Frx	
TBMsgMsg.Frm	Edit the Message
TBMsgNew.Frm	Display What's New
TBMsgNew.Frx	
TBMsgNew.Bas	
TBMsgHst.Bas	Development History
TBMsgWiz.Bas	Subroutines
TBMsgIni.Bas	Save Settings Subroutines
TBMsgWiz.Vbp	Make File
ReadMe.Txt	Current Information

Last Topic

All things being equal you should never see this screen. But alas, you have.

File: g:\tbmsgbox\help\tbmsgdef.doc

