

# MsgBox95 - An Enhanced Message Box

MsgBox95 expands upon the functions provided by Visual Basic's MsgBox. Features include:

- Supports Windows 3.x, Windows 95 and Windows NT.
- Exactly duplicates VB MsgBox functions.
- Uses the 3D effect found in newer applications.
- 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.
- Can register the message with Ctl3D.Dll to display a 3D border.
- A Wizard which creates MsgBox95 or Visual Basic code.
- A MsgBox conversion utility.

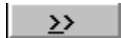
## Table of Contents

Click one of the following:

[MsgBox95 Concepts and Facilities](#)

[The MsgBox Wizard](#)

[Registration and Ordering MsgBox95](#)



Click on one of the above topics or click the Browse button to view additional documentation.

## Other MsgBox help files:

[Convert To MsgBox95](#)

[MsgBox95 Subroutines and Functions.](#)



913 Hacienda Way  
Millbrae, CA 94030  
CompuServe: 71662,205  
America Online: LRebich  
415-737-0870 Fax: 415-737-0871  
Copyright 1995

Last updated: July 23, 1995 15:00:27

Version: 3.20

Version Date: July 1995



# MsgBox95 Concepts and Facilities

Click one of the following topics:

[Sample VB and MsgBox95](#)

[MsgBox95 Components](#)

[MsgBox95.Frm and MsgBox95.Bas](#)

[MsgBox95Fun, MsgBox95Sub](#)

[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](#)

[No Separator](#)

[Register with Ctl3D.Dll](#)

[VGA Sized Messages](#)

[Bevel Width](#)

[Hide Rather Than Unload](#)

[MDI Forms.Count Routines](#)

[Prior Version Considerations](#)

[Converting from VB's MsgBox to MsgBox95](#)

[Windows 95 Considerations](#)



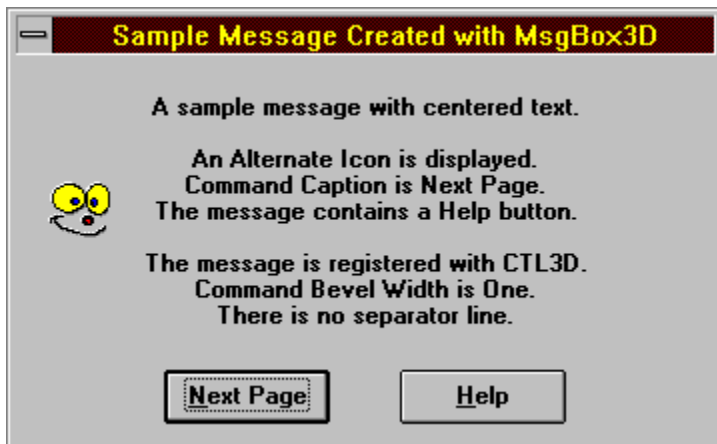
## Sample VB and MsgBox95 Messages



Windows 95 VB Message



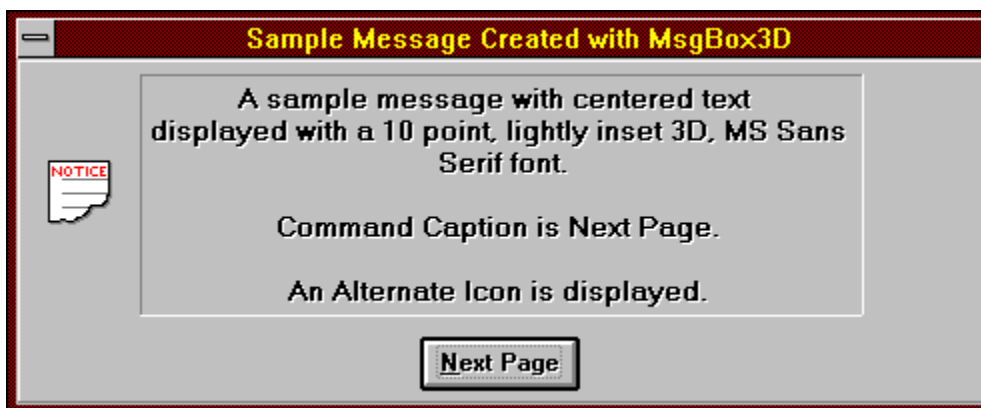
MsgBox95 - Version 3.20 and Later



3D Message - Version 3.00 and Later



3D Message - Version 2.40 and Later



3D Message - Version 2.30 and Earlier

Related Topic: [Prior Version Considerations](#)



# MsgBox95 Components

## System Components

<a href="#">MsgBox95.Frm</a>	Message Box Form
MsgBox95.Frx	Binary Data
MsgBox95.Bas	Subroutines, Functions and Globals

## Code Generation System Components

<a href="#">MsgWiz95.Exe</a>	Message "Wizard"
MsgWiz95.Mak	VB make file, used to run the system
MsgWiz95.Frm/Frx	Demo and Generate MsgBox95 messages
MsgCap95.Frm/Frx	Command captions and fonts
MsgFnt95.Frm/Frx	Message alignment and fonts
MsgHlp95.Frm/Frx	Add a Help Button
MsgIco95.Frm/Frx	Alternate icons
MsgMsg95.Frm	Edit the Message
MsgNew95.Frm/Frx	Display What's New
MsgNew95.Bas	What's New Messages
MsgWiz95.Bas	Subroutines, etc.
MsgIni95.Bas	Save Settings Subroutines
MsgHst95.Bas	History of MsgBox95 Enhancements
MsgBox95.Hlp	This Help File
MsgBSubs.Hlp	MsgBox Subs and Functions
MsgB95.Txt	Help System Constants

## Not Distributed

<a href="#">VBRun300.DLL</a>	Visual Basic Runtime Support
<a href="#">CMDialog.VBX</a>	Used for font selection
<a href="#">ThreeD.VBX</a>	Used for 3D Command effect
<a href="#">Ctl3DV2.DLL</a>	3D Border [optional]

VBRun300.Dll and ThreeD.VBX are used by MsgBox95. CMDialog.VBX is not used by MsgBox95, it is only used by MsgWiz95. Ctl3D.DLL and its newer version Ctl3DV2.DLL are only used by MsgBox95 [in Win 3.x] if they are found, see: [Registering with Ctl3D.DLL](#) for details. Ctl3D.DLL is not used with Windows 95.

## MsgConvT Components

The MsgBox conversion utility's components are described in the [MsgConvT.Hlp](#) file.



## MsgBox95.Frm and MsgBox95.Bas

You need to include `MsgBox95.Frm` and `MsgBox95.Bas` with your Visual Basic project to obtain `MsgBox95` support. First make sure you have added `ThreeD.VBX` then use VB's File, Add menu option to add the `MsgBox95` form and module.

### Calling Sequence

The `MsgBox95` calling sequence is almost identical to the VB `MsgBox` calling sequence. The subroutine is called `MsgBox95Sub`. The function is called `MsgBox95Fun`.

See [MsgBox95Fun, MsgBox95Sub](#) for details.

### Supplied Arguments

VB's `MsgBox` accepts one, two or three arguments. `MsgBox95` requires that you supply all three arguments. This is a VB limitation. See [Converting from VB MsgBox to 3D MsgBox](#).

### Special Properties

You can change the default message alignment and font properties by changing global `MsgBox95` variables. See: [Message Alignment and Fonts](#) for details.

You can also change the default command captions and font properties by changing global `MsgBox95` variables. See: [Command Captions and Fonts](#).

### Alternate Icons

You can use `MsgBox95` 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 `MsgBox95` form. See: [Help Commands](#) for details. You can display the message with no buttons. See: [No Buttons](#) for details.

### Splash Screens

You can use `MsgBox95` to create "Splash Screens." See: [Splash Screens](#) for details.

### No Titles

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

### Register with Ctl3D.DLL

You can register the message with `Ctl3D.DLL` to get a 3D type title and border. See: [Register with Ctl3D.DLL](#) for details.

### Constant.Txt

`MsgBox95` does not use the constants defined in VB's `Constant.Txt` file. Prior to version 2.90 `MsgBox95` did require these constants.



## MsgBox95Fun, MsgBox95Sub

MsgBox95 displays a message in a dialog box and waits for the user to select an option. MsgBox95Fun function returns a value indicating which option the user selected; MsgBox95Sub does not.

### Function Syntax

*return* = MsgBox95Fun(*message*, *type*, *title*)

### Subroutine Syntax

MsgBox95Sub *message*, *type*, *title*

### Remarks

**Part**      **Description**

*return*      Return code supplied by MsgBox95Fun. Integer data type.

The MsgBox95 function and statement both have these parts:

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

*type*        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:

Symbolic Constant	Value	Meaning
MB_OK	0	Display OK button only.
MB_OKCANCEL	1	Display OK and Cancel buttons.
MB_ABORTRETRYIGNORE	2	Display Abort, Retry, and Ignore buttons.
MB_YESNOCANCEL	3	Display Yes, No, and Cancel buttons.
MB_YESNO	4	Display Yes and No buttons.
MB_RETRYCANCEL	5	Display Retry and Cancel buttons.
		<i>Win 3.x Win95</i>
MB_ICONSTOP	16	 or
MB_ICONQUESTION	32	 or
MB_ICONEXCLAMATION	48	 or
MB_ICONINFORMATION	64	 or
MB_DEFBUTTON1	0	First button is default.
MB_DEFBUTTON2	256	Second button is default.
MB_DEFBUTTON3	512	Third button is default.
MB_APPLMODAL	0	Application modal. The user must respond to the message box before continuing work in the current application.
MB_SYSTEMMODAL	4096	System modal. The user must respond to the message box before continuing work in any application.



The first group of values (1-5) 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 type, use only one number from each group. If omitted, the default value for type is 0.

*title* String expression displayed in the title bar of the dialog box. If you omit the argument title, MsgBox95 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 MsgBox95. See [No Title](#) for details.

## Displaying the Message

MsgBox95 displays as many characters as will fit in a LabelBox. Longer messages are truncated. 3D messages are limited to 255 characters, a limitation of the ThreeD.Vbx Panel control.

MsgBox95 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, Chr\$(13)] or a linefeed [ANSI 10, Chr\$(10)] before the first character of the text that is to begin each new line.

## Returned Value

The MsgBox95 function *return* value indicates which button has been selected, as shown in the following table:

Symbolic constant	Value	Meaning
IDOK	1	OK button selected.
IDCANCEL	2	Cancel button selected.
IDABORT	3	Abort button selected.
IDRETRY	4	Retry button selected.
IDIGNORE	5	Ignore button selected.
IDYES	6	Yes button selected.
IDNO	7	No button selected.

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

## Note

Note: Symbolic constants for all message box definitions can be found in the Visual Basic file [Constant.Txt](#). When placed in any module in a project, the symbolic names can be used in all your form and code modules.

Note: Most of this topic has been copied from Microsoft's Visual Basic Help file. It has been edited, where necessary, to describe MsgBox95 capabilities.



## Message Alignment and Fonts

Message alignment and font selection information can be passed to `MsgBox95` in the following global variables:

```
Global MB3D_Alignment As Integer
Global MB3D_FontName As String
Global MB3D_FontSize As Single
Global MB3D_FontNotBold As Integer
Global MB3D_FontItalic As Integer
Global MB3D_Font3D As Integer
```

Move information into the variables before calling `MsgBox95`. The variables are never changed by `MsgBox95`. 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, non-bold, italic, 3D raised, light font, 12 point, "Arial" message to be displayed is:

```
.....
MB3D_Alignment = 2           'center
MB3D_FontName = "Arial"
MB3D_FontSize = 12
MB3D_FontNotBold = True
MB3D_FontItalic = True
MB3D_Font3D = SS_Font3D_Raised_Light
.....
MsgBox95Sub "A Message", MB_Icon..., "A Title"
.....
```

### Defaults

The global variables are initialized with their "null" values. If you don't change them then `MsgBox95` uses the "design time" alignment and font properties ["MS Sans Serif", bold, non-italic, 8.25 points, no 3D effect, with left alignment].

### Constant.Txt

Text alignment and 3D font values are defined in VB's help file. Standard global settings are defined as constants in VB's `Constant.Txt` file. `MsgBox95` does not use the constants. You will want to include them in your code. They are:

#### Alignment Constants:

```
Global Const LEFT_JUSTIFY = 0 ' 0 - Left Justify
Global Const RIGHT_JUSTIFY = 1 ' 1 - Right Justify
Global Const CENTER = 2 ' 2 - Center
```

#### 3D Font Constants:

```
Global Const SS_FONT3D_NONE = 0
Global Const SS_FONT3D_RAISED_LIGHT = 1
Global Const SS_FONT3D_RAISED_HEAVY = 2
Global Const SS_FONT3D_INSET_LIGHT = 3
Global Const SS_FONT3D_INSET_HEAVY = 4
```

### 3D Font Considerations

`MsgBox95` uses a standard VB Label to display non-3D messages. A Label can display a very large number of characters. `MsgBox95` uses the `SSPanel` control [part of `ThreeD.Vbx`] to display 3D messages. The `SSPanel` will only display 255 characters. `MsgBox95` issues a warning message if the message you are attempting to display is too large for the `SSPanel`. If

too large, the message is displayed with a non-3D font.



## Command Captions and Fonts

Command captions and font selection information can be passed to `MsgBox95` 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_FontNotBold As Integer
Global MB3D_Commands_FontItalic As Integer
Global MB3D_Commands_Font3D As Integer
Global MB3D_Commands_No3D As Integer
```

Move information into the variables before calling `MsgBox95`. The variables are never changed by `MsgBox95`. 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", inset, light command called "Next Page" to be displayed is:

```
.....
MB3D_UD_OK = "&Next Page"
MB3D_Commands_FontName = "Arial"
MB3D_Commands_FontSize = 12
MB3D_Commands_FontNotBold = True
MB3D_Commands_FontItalic = True
MB3D_Commands_Font3D = SS_FONT3D_INSET_LIGHT
.....
MsgBox95Sub "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 `MsgBox95` 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, except `Font3D`, are initialized with their "null" values. If you don't change them then `MsgBox95` uses the "design time" captions and font properties ["MS Sans Serif", bold, non-italic, 8.25 points]. `Font3D` is initialized as Raised-Heavy to retain compatibility with earlier versions of `MsgBox95`.

## No 3D Font

If you do not want a command button 3D font then use the following code in you application:

```
MB3D_Commands_No3D = True
```

If "No3D" is true it overrides the "Font3D" setting. The "Font3D" setting is ignored if "No3D" is true.

## Constant.Txt

3D font values are defined in VB's help file. Standard global settings are defined as constants in VB's Constant.Txt file. MsgBox95 does not use the constants. You will want to include them in your code. They are:

### 3D Font Constants:

```
Global Const SS_FONT3D_RAISED_LIGHT = 1
```

```
Global Const SS_FONT3D_RAISED_HEAVY = 2
```

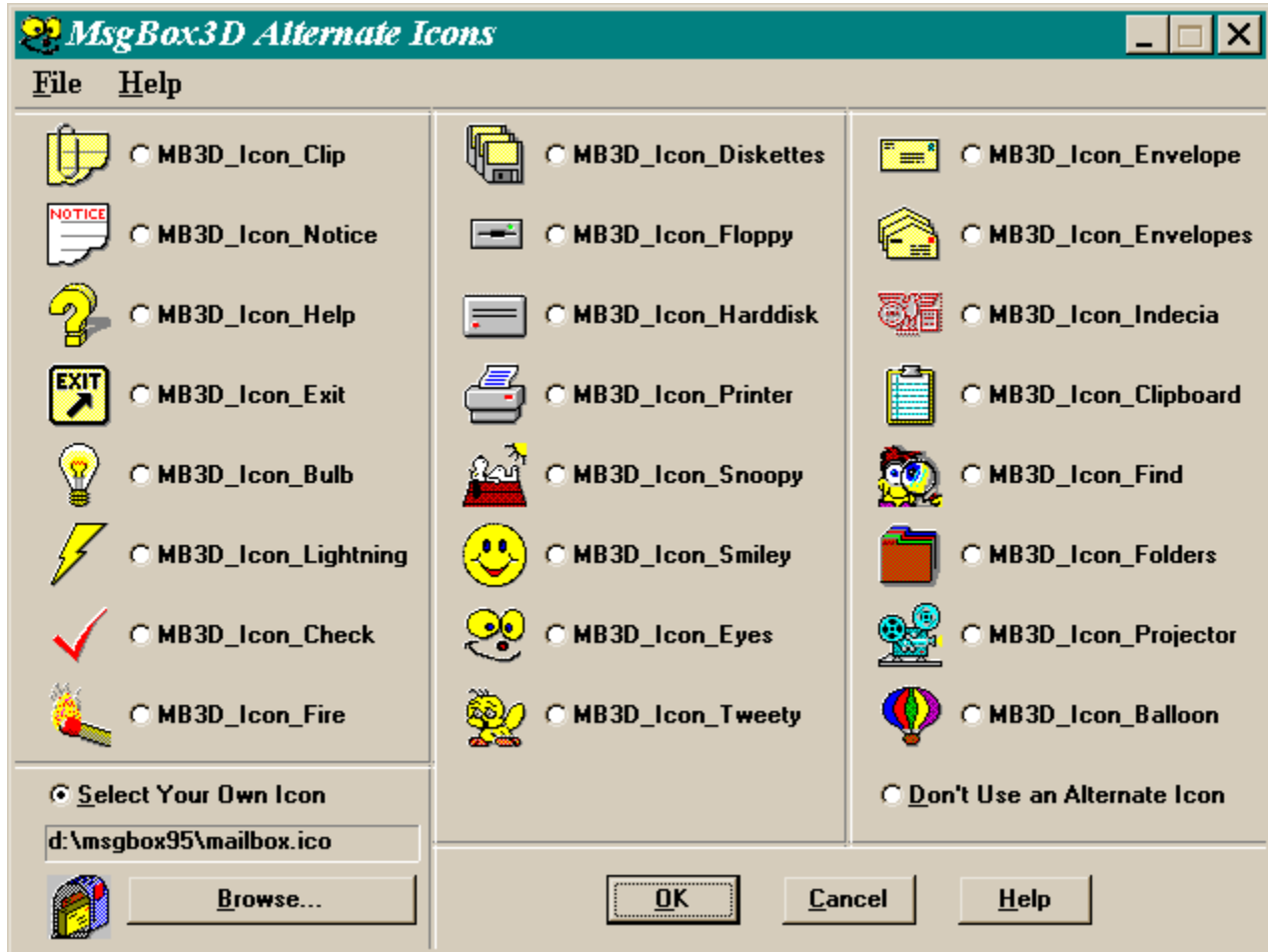
```
Global Const SS_FONT3D_INSET_LIGHT = 3
```

```
Global Const SS_FONT3D_INSET_HEAVY = 4
```



## Alternate Icons

MsgBox95 will display 24 alternate icons in addition to the 4 standard VB icons. And MsgBox95 will also use an icon specified by the icon's fully qualified file name. The 24 alternate icons are displayed on this screen. There are no **hot spots** on this screen.



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



## Selecting An Alternate Icon

Before calling MsgBox95 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 MsgBox95 to display the specified icon with your message. For example:

```
MB3D_Icon_FileName = "d:\MsgBox95\mailbox.ico"
```

If MsgBox95 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** - MsgBox95 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 MsgBox95 form [form3DMsgBox]. The image controls are indexed. The index number directly corresponds to a **constant** defined in MsgBox95.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, MsgBox95 was loaded and unloaded each time it was used. If you still load rather than hide MsgBox95 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**, MsgBox95 loads the VB image control [ImageUserLoaded] with the icon each time MsgBox95 is activated. Statements similar to these are used:

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





## Saving and Restoring Globals

**MsgBox95** 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  
  
MsgBox95Sub 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 MsgBox95. Be sure to dimension the array before calling the save subroutine.**



## Ctrl-P Prints the Message

Press **Ctrl-P** to send a MsgBox95 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. MsgBox95 uses Visual Basic's **PrintForm** method. PrintForm does not print the Title or the Command Buttons.

### History

This feature was implemented in MsgBox95 version 3.10, June 1995.

## Message Location

By default the MsgBox95 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
MsgBox95Sub "Msg", ...
```

MB3D\_Zone uses one parameter - the name of a form. Another example:

```
MB3D_Zone MyMDIForm
MsgBox95Sub "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 MsgBox95 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
MsgBox95Sub "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 MsgBox95 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 `MsgWiz95` application. See: [MsgWiz95.Exe](#).

## History

This feature was implemented in `MsgBox95` Version 2.46, November 1994.



## Splash Screens

MsgBox95 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 MsgBox95 will simulate the user pressing the first button on the message. The value is specified in milliseconds. For example:

```
MB3D_Splash = 1000
```

causes MsgBox95 to simulate the user pressing the message's first command button after a one second delay.

MsgWiz95's Help, About [Splash] menu demonstrates this MsgBox95 feature.

**Visual Basic Timer** - MsgBox95 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 MsgBox95 form if **MB3D\_HelpCommand** is not zero. Four **global variables** are specified. An example:

```
MB3D_HelpCommand = &H105      'Help_PartialKey
MB3D_HelpFile = "d:\MsgBox95\MsgBox95.hlp"
MB3D_HelpData = "about"      'Search String
MB3D_HelpCaption = "&Learn"  'Alternate Command Caption
```

### MB3D\_HelpCommand

Any valid WinHelp command can be specified. See the **Visual Basic Constant.Txt** file for a list of valid commands. 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**. MsgBox95 uses Window's file search logic to find the file. If the file is not found MsgBox95 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 and 3D effect, specified for other MsgBox95 commands, is applied to the Help button. In other words; all MsgBox95 command buttons are made to look the same. See: [Command Captions and Fonts](#).

## Using MsgBox95's Help in Your Application

You can use the MsgBox95 help launcher in your application. The process is fully described in [Sub MB3D\\_DoHelp](#) in file MsgBSubs.Hlp.



## Mouse "Snap-To" Default Button

MsgBox95 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 MsgBox95.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 MsgBox95 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
MsgBox95Sub 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 = MsgBox95Fun(Msg, ..., Ttl)
If Rtn = IDCANCEL 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", IDNO is returned if the user presses "Enter". Pressing Esc always returns IDCANCEL even if no cancel button was specified.



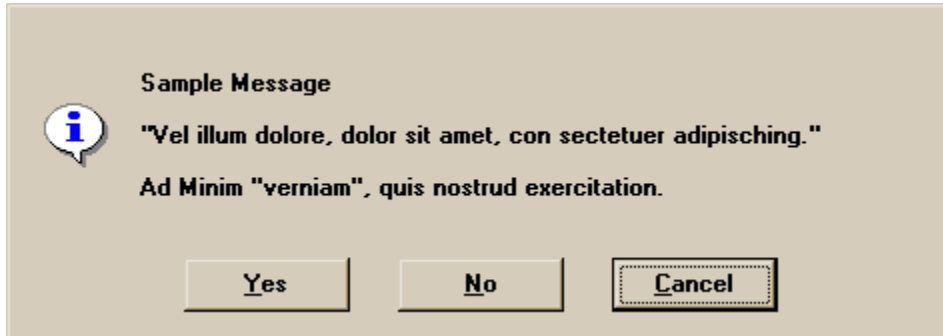
## No Title

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 MsgBox95. For example:

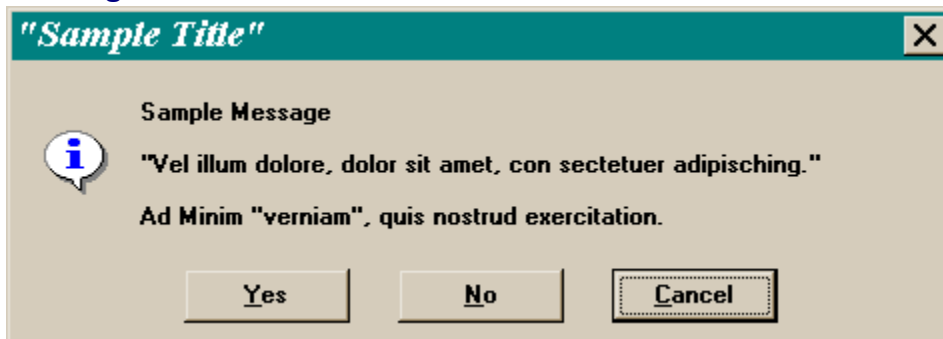
```
MB3D_NoTitle = True
MsgBox95Sub ....
```

This variable is saved and restored with [MB3D Globals Save/Restore](#).

## Message without a Title



## Message with a Title



## Move Using Drag

You can move the message by dragging the top part of the message, except the icon. Hold down the left mouse button then drag the message to the desired location. Drop the message by releasing the mouse button.

## Mouse Click

MsgBox95 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

MsgBox95 uses Windows API SetWindowLong to change the style of the message box. See subroutine MB3D\_SetTitleAndCtl3D in module MsgBox95.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 MsgBox95 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].

**THE  
EDGE**

## Register with Ctl3D.Dll

With Windows 3.x, MsgBox95 uses function Ctl3dSubclassDlgEx provided in newer versions of [Ctl3D.DLL](#) or [Ctl3DV2.DLL](#) to add a 3D border effect if you set MB3D\_UseCtl3D to true.

Sample calling sequence:

```
MB3D_UseCtl3D = True
MsgBox95Sub ....
```

### Sample Registered Message



### Sample Unregistered Message



## Technical Information - Windows 3.x

Before issuing the first message, MsgBox95 attempts to find [Ctl3DV2.DLL](#) then [Ctl3D.DLL](#). It then uses function Ctl3DGetVer to get their version number. The Ctl3dSubclassDlgEx function was added in version 2.0. Since any Ctl3DV2.DLL is, by definition, greater than version 2.0 it is used first. If MsgBox95 can't find Ctl3DV2.DLL it uses Ctl3D.DLL if its version number is 2.0 or greater. If it can't find either or if Ctl3D.DLL is an old version then MB3D\_UseCtl3D is "forced" to False.

Only messages with a title are registered. If you have set [MB3D\\_NoTitle](#) to true then the message is not registered with the DLL.

You may occasionally encounter a GPF while developing an application if you do not unload the message form before ending your application.

This variable is saved and restored with [MB3D\\_Globals\\_Save/Restore](#).

## Windows 95

This feature only applies to messages displayed in Windows 3.10. It is ignored if the message is displayed in Windows 95.

## Windows NT

This feature is not used if the message is displayed in Windows NT. There is no way to prevent a GPF when your application terminates in Windows NT.

## History

This feature was implemented in MsgBox95 Version 2.66, March 1995. Ctl3D Version checking was added in Version 2.71, March 1995 [thanks to user Dallas Bridge, 100237,3155].

Windows 95 and Windows NT testing was performed while developing Version 3.20, July 1995.



## No Separator

The message can be displayed with or without a separator line between the message and the command buttons. With a separator:



Without a separator:



## MB3D\_NoSeparator

Set MB3D\_NoSeparator to True to display the message without a separator line. For example:

```
MB3D_NoSeparator = True
```

This variable is saved and restored with [MB3D Globals Save/Restore](#).

## Windows 95

The separator is never displayed with messages issued in Windows 95.

## History

This feature was implemented in MsgBox95 Version 2.84, April 1995. Thanks to Jeffrey McManus [72103,3263] for the suggestion. Automatic removal of the separator while displaying the message in Windows 95 began in Version 3.20, July 1995.



## VGA Sized Messages

MsgBox95 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 MsgBox95. For example:

```
MB3D_VGA_Messages = True
MsgBox95Sub ....
```

This variable is saved and restored with [MB3D\\_Globals\\_Save/Restore](#).

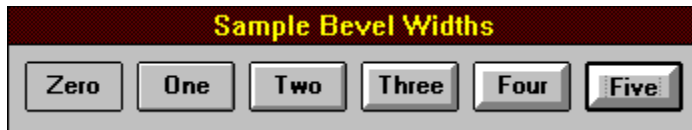
## History

This feature was implemented in MsgBox95 Version 2.52, January 1995.



## Bevel Width

The command button's bevel width can be changed by setting global variable MB3D\_BevelWidth to a value from 1 to 10 and a negative number. Default is 2. Samples:



Setting the bevel width to one provides a Windows 95 command button appearance. The bevel width is set to 1 in the following sample:

```
MB3D_BevelWidth = 1
MsgBox95Sub ...
```

This variable is saved and restored with [MB3D\\_Globals\\_Save/Restore](#).

## Windows 95

This feature is not needed and its setting is ignored when the message is displayed in the Windows 95 environment.

## Technical

If MB3D\_BevelWidth = 0 then the default width [2] is used. To display the command button with a zero bevel width set MB3D\_BevelWidth to a negative number, for example minus one:

```
MB3D_BevelWidth = -1 'Set BevelWidth to Zero
```

This implementation preserves compatibility with earlier versions of MsgBox95.

## History

The feature was implemented in MsgBox95 Version 2.91, May 1995. A suggestion from Bruce Nordstrand [100352,176].





## Hide Rather Than Unload

To dramatically improve performance you can set **MB3D\_UnloadAfterDisplay** to False to cause the MsgBox95 form to be hidden after use rather than unloaded. For example:

```
Global Const MB3D_UnloadAfterDisplay = False
```

This constant is defined in module MsgBox95.Bas. The default setting is "False."

## MB3D\_Unload

You can force the MsgBox95 form to be unloaded by calling subroutine MB3D\_Unload. To demonstrate this feature, a MsgWiz95.Exe file menu item calls this subroutine. Use it to compare MsgBox95 performance, force the form to be unloaded by using File, Unload then display a MsgBox95 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 MsgBox95 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 MsgBox95 program logic was moved from the Form [MsgBox95.Frm] to the Module [MsgBox95.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 MsgBox95 system modal message is *always* unloaded - never hidden.



## MDI Forms.Count Routines

You may find the following routines useful if you hide [rather than unload] MsgBox95 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
  MDIAllUnloadSkip:
  Next
  Exit Sub
MDIAllChildUnloadError:
  Resume MDIAllUnloadSkip
```

End Sub



## Prior Version Considerations

Prior to version 2.34 the message box was displayed in an [outline box](#). If you want to continue to use the "old" style then set the following constant to True:

```
Global Const MB3D_233StyleMessage = True
```

The constant is defined in `MsgBox95.Bas`. The distributed source code value is "False".



# Converting from VB's MsgBox to MsgBox95

## Manual or Automatic

You can manually convert your VB projects or you can use the supplied Conversion Utility [MsgConv]. The manual process is described below. Refer to the [MsgConv.Hlp](#) file for instruction on using the conversion utility. MsgConv was first distributed with MsgBox95 3.00, May 1995.

## Manual Conversion

Convert VB MsgBox statements to **MsgBox95Sub** subroutines. Convert VB MsgBox functions to **MsgBox95Fun** functions. For example the function:

```
RtnCode = MsgBox(TheMessage, MB_..., TheTitle)
```

when converted becomes:

```
RtnCode = MsgBox95Fun(TheMessage, MB_..., TheTitle)
```

## Add Files

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

<a href="#">MsgBox95.Frm</a>	Message Box Form
<a href="#">MsgBox95.Bas</a>	Subroutines, Functions and Globals
<a href="#">ThreeD.VBX</a>	Used for 3D effect

Be sure to add ThreeD.VBX before adding MsgBox95.Frm. Otherwise, VB produces errors similar to:

```
Class SSCommand of control Command3DCommand was not a loaded control class.  
The property name Caption in Command3DCommand is invalid.  
The property name Font3D in Command3DCommand is invalid.
```

## SS3D.VBX

Sheridan's SS3D.VBX was enhanced and renamed ThreeD.VBX when distributed with Visual Basic. You can continue to use SS3D.VBX instead of ThreeD.VBX. You can not add both SS3D.VBX and ThreeD.VBX to a project. Use one or the other. If you convert from SS3D.VBX to ThreeD.VBX start a new project then "manually" add each form, module and VBX. Add VBX's first, then modules and forms. Thanks go to Kip Johnson, 76451,3660, for relaying this information.

## 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.

## Considerations

**Same Number of Arguments** - The *type* and *title* are not required with VB's MsgBox, they are required with MsgBox95. A VB limitation - Argument Count Mismatch.

**Data Type** - The data type of the parameters passed to or returned from VB's MsgBox can be *variant*. They must be either *string* or *integer* with MsgBox95. A VB limitation - Parameter Type Mismatch.



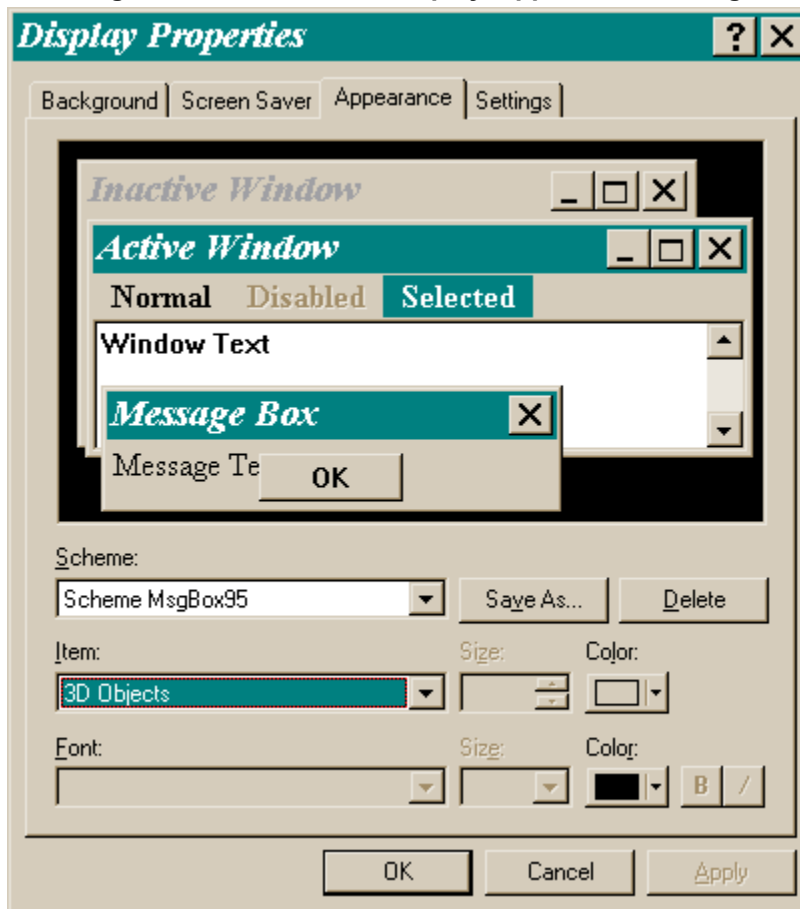
## Windows 95 Considerations

MsgBox95 is designed to perform in both the Windows 3.x and Windows 95 environment. Messages you design under Windows 3.x or Windows 95 are displayed using the characteristics of the host operating system. That is: if your user runs your application in Windows 3.x then converts to Windows 95 your application's message will take on the characteristics of Windows 95. Your message do not have to be changed. MsgBox95 automatically makes the following conversion for you:

### Command Buttons

MsgBox95.Frm contains both the ThreeD.VBX [Sheridan] command button and the standard VB command button. The Sheridan button is used with Windows 3.x and the standard VB command button with Windows 95. The Sheridan button does not have a BackColor property - it always uses light gray.


Standard command buttons are displayed with the Windows 95 3D Objects color as defined using the Control Panel's, Display, Appearance setting. For example:



### Use of Ctl3D.DLL

Windows 95 automatically incorporates a 3D look. If a message is [registered with Ctl3D.DLL](#) that registration is ignored if the message is issued in the Windows 95 environment.

### Standard Message Icons

The stop, exclamation, information and question icons are automatically changed to the new Windows 95 message icons. For example a message issued in Windows 3.1 using  is

changed to use



in Windows 95.

## Message Beep

Standard Windows 95 messages play the Message Beep sound defined in Control Panel Sound settings. MsgBox95 plays the same sound. Messages using [Alternate Icons](#) are displayed without sound.

## Separator

The message separator is never displayed in Windows 95. It is optional with Windows 3.x. See: [MB3D\\_NoSeparator](#) for details.

## 3D Fonts

Command button fonts are no longer displayed in Windows 95. Commands with 3D fonts can still be displayed in Windows 3.x. Message fonts can still be displayed in Windows 95 - although most Windows 95 applications do not display messages using a 3D font.

## Bevel Width

MB3D\_BevelWidth is ignored in Windows 95. See [Bevel Width](#) for details.

## Interim Version

MsgBox95 3.20 is an interim release of the product. It supports both the Windows 3.x and Windows 95 GUI. It is intended that a future version will drop support for Windows 3.x after a significant number of users convert to Windows 95. There is a significant amount of duplicated source code needed to support both environments. Much of that duplicated code will be removed when a "final" version of MsgBox95 is released in September or October 1995.





## The MsgBox Wizard

Click one of the following topics:

[MsgWiz95.Exe](#)

[Message Alignment And Fonts](#)

[Command Captions and Fonts](#)

[Selecting a Font](#)

[Selecting an Alternate Icon](#)

[Selecting a Help File](#)

[Generating MsgBox95 Source Code](#)

[MsgWiz95 Settings are Saved](#)

[Displaying What's New](#)







## MsgWiz95.Exe - The MsgBox Wizard

MsgWiz95 is used to generate Visual Basic source code that you can copy into your project. The source code is written to the Clipboard. MsgWiz95 also demonstrates the similarities and differences between VB's MsgBox and MsgBox95. You may wish to review VB's help topic [MsgBox Function, MsgBox Statement](#). MsgBox95 replicates the functionality of VB's MsgBox. MsgWiz95 is written in VB 3.0. It uses [VBRUN300.DLL](#). Click on the following form for additional information about each section of this screen. There are 16 [hot spots](#) on this screen.

Select desired option buttons then click the VB MsgBox or MsgBox95 commands to display the standard VB or MsgBox95 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 MsgBox95 message's alignment and font properties with this form. Click on the following form for additional information about each section. There are 13 **hot spots** on this screen.

**Select a Message Font and Alignment**

File Alignment 3D Font Help

Arial, 12, Bold  
Inset Light

Click the Left, Center or Right option to change the message text alignment.

Click Raised, Inset, Light or Heavy to display a 3D font.

Click the 'Select a Different Font...' command to change the message's font properties.

Left  Center  Right

None Raised Inset Light Heavy

Select a Different Message Font...

Default OK Cancel Help

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





## Command Captions and Fonts

You can change MsgBox95 command's captions and font properties with this form. Click on the following form for additional information about each section. There are 13 hot spots on this screen.

Default	Replacement	Enter Replacement Here
OK	OK	&OK
Cancel	Abbrechen	&Abbrechen
Abort	Abbrechen	&Abbrechen
Retry	Wiederholen	&Wiederholen
Ignore	Ignorieren	&Ignorieren
Yes	Ja	&Ja
No	Nein	&Nein

None   Raised   Inset   Light   Heavy

Select a Different Caption Font...

Default   OK   Cancel   Help

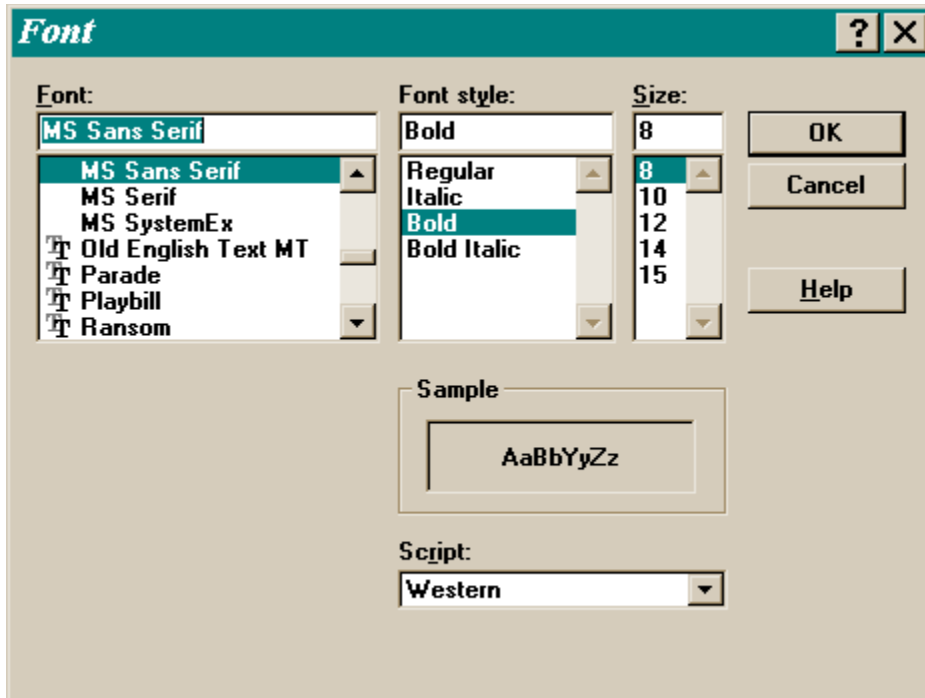
This screen is centered within a zone defined using [MB3D\\_Zone](#).





## Selecting a Font

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

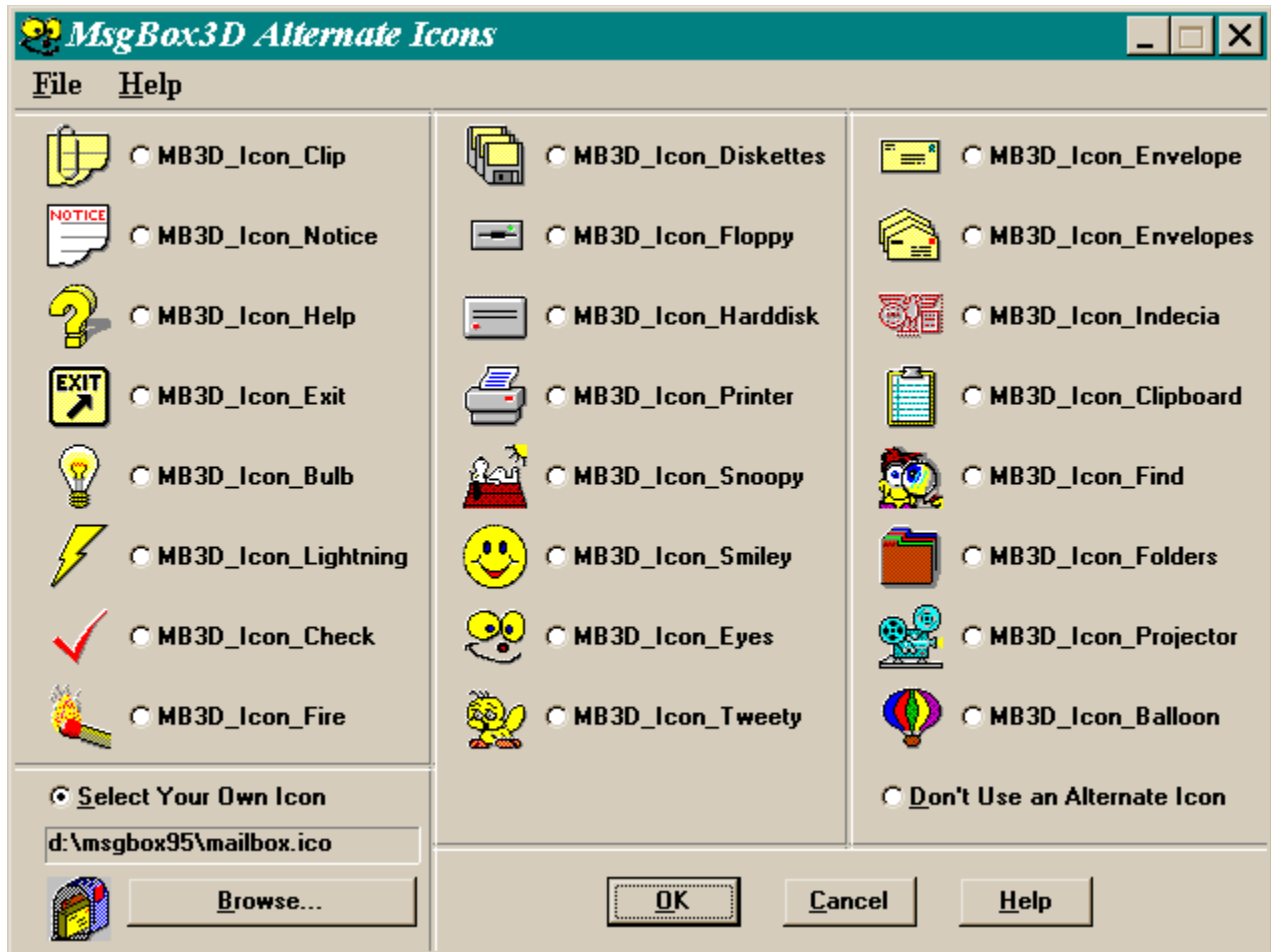


In Windows 3.x, MsgWiz95 will use CTL3D.DLL or CTL3DV2.DLL, if found on your system, to display common dialog screens with a 3D effect.



## Selecting an Alternate Icon

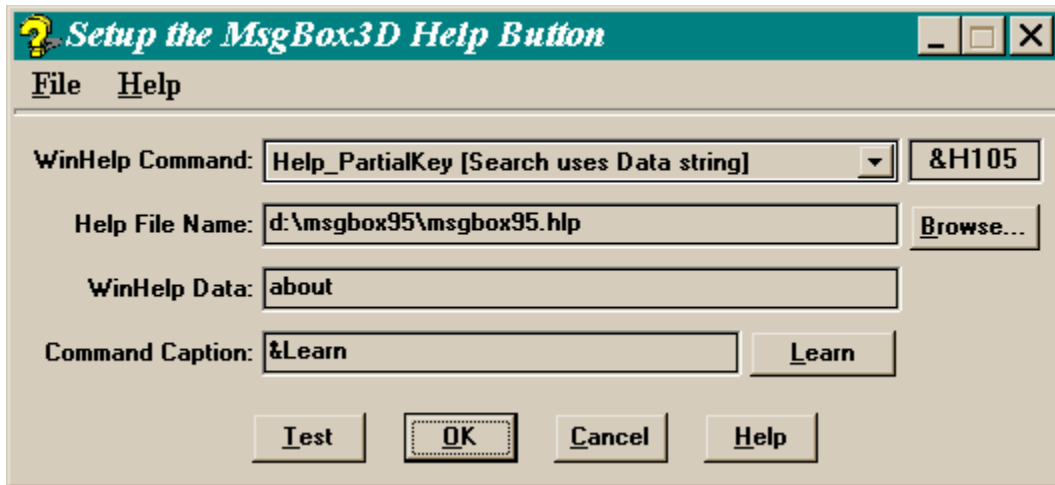
The following screen is displayed when you click the "Use an Alternate Icon" option or menu item File, Select an Alternate Icon. Click on the following form for additional information about each section. There are 6 hot spots on this screen.



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.



**Setup the MsgBox3D Help Button**

File Help

WinHelp Command: Help\_PartialKey [Search uses Data string] &H105

Help File Name: d:\msgbox95\msgbox95.hlp Browse...

WinHelp Data: about

Command Caption: &Learn Learn

Test OK Cancel Help

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



## Generating MsgBox95 Source Code

If File menu item **Write Generated Code to the Clipboard** is checked, MsgWiz95 will generate the Visual Basic code used to create the sample message. The code is written to the clipboard. You could paste the code into your VB application.

### Sample Generated Code

MsgWiz95 generated the following code. VB comments have been eliminated to make the code easier to read.

```
'09/07/1994-10:41:07, Code generated by MSGWIZ95
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 & "Sample Message:" & Chr$(13)
MB3D_Msg = MB3D_Msg & Chr$(13)
MB3D_Msg = MB3D_Msg & Chr$(34)
MB3D_Msg = MB3D_Msg & "Vel illum dolore, sectetuer adipisching."
MB3D_Msg = MB3D_Msg & Chr$(34)
MB3D_Msg = MB3D_Msg & Chr$(13)
MB3D_Msg = MB3D_Msg & Chr$(13)
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 & Chr$(34)
MB3D_Title = MB3D_Title & "Sample Title"
MB3D_Title = MB3D_Title & Chr$(34)
MB3D_Type = MB3D_Type + 3      'MB_YesNoCancel
MB3D_Type = MB3D_Type + 64   'MB_IconInformation
MB3D_Type = MB3D_Type + 512  'MB_DefButton3
MB3D_ReturnCode = MsgBox95Fun (MB3D_Msg, MB3D_Type, MB3D_Title)
MB3D Globals Restore MB3D_SaveArray()
If MB3D_ReturnCode = 6 Then  'IDYes
End If
If MB3D_ReturnCode = 7 Then  'IDNo
End If
If MB3D_ReturnCode = 2 Then  'IDCancel
End If
```

Beginning with version 2.81, April 1995, output is created for VB MsgBox as well as MsgBox95.



## **MsgWiz95 Settings are Saved**

When you exit MsgWiz95 options and settings you selected are saved in MsgWiz95.Ini. These settings are restored the next time the program is executed.

A sample Ini file:

```
; MsgWiz95.Ini - MsgWiz95 Initialization File
; 95/04/02 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_FontNotBold=False
MB3D_FontItalic=False
MB3D_Font3D=SS_Font3D_None

[Saved Command Button Font Information]
MB3D_Commands_FontName=
MB3D_Commands_FontSize=0
MB3D_Commands_FontNotBold=False
MB3D_Commands_FontItalic=False
MB3D_Commands_Font3D=SS_Font3D_None
MB3D_Commands_No3D=True

[Saved Help Information]
MB3D_HelpCommand=261
MB3D_HelpFile=d:\space.020\MsgBox95\MsgBox95.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
MB3D_UseCtl3D=False
```

```
[Saved MsgWiz95 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 stored in MsgWiz95's 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 MsgWiz95 location was implemented in MsgBox95 Version 2.47, November 1994.  
Saving selected settings was implemented in version 2.74, March 1995.



## Displaying What's New

You can display the historical record of MsgBox95 development. Click menu item What's New... The following dialog is displayed the first time you run MsgWiz95:



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

Click the What's New... menu item to display this message again the next time you begin MsgWiz95.



# Registration and Ordering MsgBox95

Click one of the following topics:

[Registration and Ordering MsgBox95](#)

[Registered User's License](#)

[Upgrade Policy](#)

[About MsgBox95](#)

[Other Products](#)

[Update Information](#)



# Registration and Ordering MsgBox95

You can order MsgBox95'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 a America Online message to LRebich, 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.

## MsgBox95 Components

Click here to view [MsgBox95 Source Components](#).

## Ship MsgBox95 To

Name: \_\_\_\_\_  
Company: \_\_\_\_\_  
Address: \_\_\_\_\_  
City/State/Code \_\_\_\_\_  
Country \_\_\_\_\_  
Phone: \_\_\_\_\_  
CompuServe: \_\_\_\_\_ or,  
America Online: \_\_\_\_\_

## 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  
If mailed a 3 1/2 inch diskette is used.

## 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:



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

**Last updated: July 23, 1995 15:00:28**

**Version: 3.20**

**Version Date: July 1995**

# MsgBox95 - Registered User License

**You should carefully read the following terms and conditions before using MsgBox95. 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 MsgBox95 (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 [MsgBox95.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.**



**913 Hacienda Way  
Millbrae, CA 94030**

**CompuServe: 71662,205  
America Online: LRebich  
415-737-0870  
Fax: 415-737-0871  
Copyright 1995**

# Upgrade Policy

## Registered 3.00 Users

Registered users of version 3.00 or later can order MsgBox95 at no charge by agreeing to accept the CompuServe transmission charges - usually about \$1.00. Send a message to The Bridge or use the [Order Form](#).

## Registered Users Prior to 3.00

MsgBox95 is available for \$9.95 to registered users of earlier versions. There is no shipping charge. Please send a message to The Bridge [CompuServe: 71662,205 or America Online: LRebich] noting your VISA or MasterCard number, with expiration date. Or send a check to:



913 Hacienda Way  
Millbrae, CA 94030

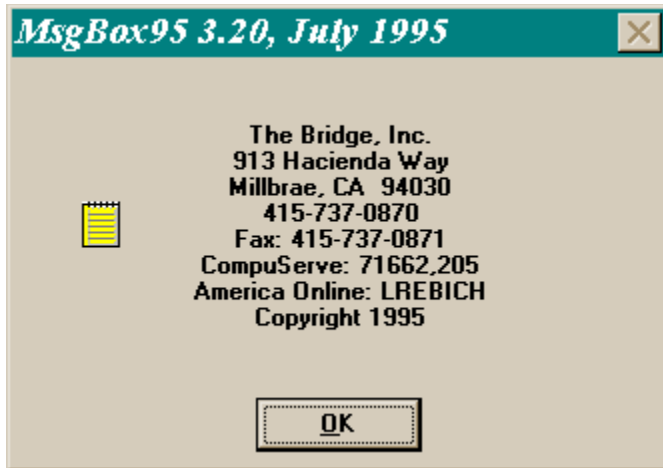
Or fax the [Order Form](#) to 415-737-0871.





## About MsgBox95

Click the right mouse button on a MsgBox95 Icon, or press F12, while displaying a MsgBox95 message to display this "About" box:



The message can also be displayed from the  
MsgWiz95's help menu [Help, About].



## Other Products

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

### HolCal - A Holiday Calendar

A full function holiday calendar. Use it as a stand alone calendar and while minimized as a clock. Or use it to communicate with your application via DDE. Holiday information is stored in an Access database. Source code is available for \$19.95. MSBasic forum file name: HolCal.Zip. Version 2.0 released March 1995. Over 300 downloads.

### DemoCt3D

Demonstrates the API's needed to call Ctl3D.DLL. Freeware. File name: DmoC3D.Zip. Over 1230 people have downloaded this demo.

### 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. About 150 downloads.

### Shade

A demo program that shows how to shade a form - similar to the Window's Setup screen. Freeware. File name: Shade.Zip. Over 500 downloads.

### SetTab

A demo program that shows how to set tabs in a text box. Freeware. File name: SetTab.Zip. Over 740 downloads.

### TstLay

Two functions that will save and restore a TrueGrid layout in a binary file. Freeware. File name: TstLay.Zip. Over 270 downloads.

### 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. Over 130 downloads.



# Update Information

Last updated: July 23, 1995 15:00:28

Version: 3.20

Version Date: July 1995



## 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 MsgBox95Sub

A typical MsgBox95Sub calling sequence:

MsgBox95Sub message, type, title

For example:

MsgBox95Sub "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 MsgBox95Fun

A typical MsgBox95Fun calling sequence:

```
RtnCode = MsgBox95Fun (message, type, title)
```

For example:

```
RtnCode = MsgBox95Fun ("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]**.

# MsgBox

## ' MsgBox Constants

Global Const MB_OK = 0	' OK button only
Global Const MB_OKCANCEL = 1	' OK and Cancel buttons
Global Const MB_ABORTRETRYIGNORE = 2	' Abort, Retry, and Ignore buttons
Global Const MB_YESNOCANCEL = 3	' Yes, No, and Cancel buttons
Global Const MB_YESNO = 4	' Yes and No buttons
Global Const MB_RETRYCANCEL = 5	' Retry and Cancel buttons
Global Const MB_ICONSTOP = 16	' Critical message
Global Const MB_ICONQUESTION = 32	' Warning query
Global Const MB_ICONEXCLAMATION = 48	' Warning message
Global Const MB_ICONINFORMATION = 64	' Information message
Global Const MB_DEFBUTTON1 = 0	' First button is default
Global Const MB_DEFBUTTON2 = 256	' Second button is default
Global Const MB_DEFBUTTON3 = 512	' Third button is default
Global Const MB_APPLMODAL = 0	' Application Modal Message Box
Global Const MB_SYSTEMMODAL = 4096	' System Modal

## ' MsgBox return values

Global Const IDOK = 1	' OK button pressed
Global Const IDCANCEL = 2	' Cancel button pressed
Global Const IDABORT = 3	' Abort button pressed
Global Const IDRETRY = 4	' Retry button pressed
Global Const IDIGNORE = 5	' Ignore button pressed
Global Const IDYES = 6	' Yes button pressed
Global Const IDNO = 7	' No button pressed

## **VBRun300.DLL - VB Dynamic Link Library**

**MsgWiz95 is written in Visual Basic. VBRun300.DLL is needed to run the it. Make sure the Windows/System directory contains a copy of this Dynamic Link Library.**



## ThreeD.VBX - Three Dimensional Controls

ThreeD.VBX is distributed with Visual Basic. MsgBox95 uses its three dimensional command support. Add this VBX to your project if you plan to use MsgBox95.

ThreeD.VBX was developed by Sheridan Software. The predecessor product was called SS3D.VBX. You can continue to use that VBX in your project. You can not use both in a project. See: [Converting from VB's MsgBox to MsgBox95](#) for details.

## **CMDialog.VBX - Common Dialog Control**

The Common Dialog control provides support for a number of commonly used dialog box functions. They are: Open a File, Save a File, Print, select Colors and select Fonts.

CMDialog.VBX interfaces with Window's Dynamic Link Library component CommDlg.DLL.

Make sure both are available in your Window's, System directory.

MsgWiz95 uses CMDialog.VBX to select fonts and locate an alternate icon. It is not used by MsgBox95.

MsgWiz95 attempts to locate Ctl3D.DLL or Ctl3DV2.DLL. If one is found on your system then the common dialog is displayed with a 3D effect.

## **Ctl3D.DLL and Ctl3DV2.DLL**

These dynamic link library components are provided with Windows. If used, they provide the standard 3D effect used with many Windows application.

MsgBox95 first uses the newer version [Ctl3DV2] if found then uses the older version if available. MsgBox95 verifies that Ctl3D version 2.0 or greater is installed.

## 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 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.

MsgBox95 uses Window's API SetSysModalWindow to set a system wide modal message. There are a few differences between the VB MsgBox and MsgBox95 message displayed in system modal form. Differences:

### VB MsgBox

- No Title or Control Box in some cases.
- No border in some cases.

### MsgBox95

- If MB3D\_NoTitle=False, always has a Title and Control Box.
- Always has a border.

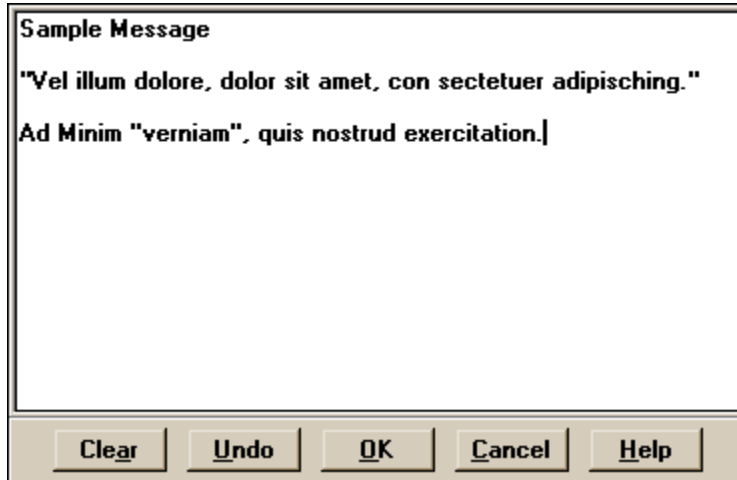
## **Subroutine or Function**

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

The image shows a rectangular dialog box with a double-line border. At the top left, the text "Sample Message" is displayed. Below this, there is a text area containing two lines of Latin placeholder text: "Vel illum dolore, dolor sit amet, con sectetuer adipiscing." and "Ad Minim "verniam", quis nostrud exercitation.]. At the bottom of the dialog box, there is a horizontal row of five buttons: "Clear", "Undo", "OK", "Cancel", and "Help". Each button has a small icon to its left and is styled with a light beige background and a thin border.

**History** - Undo was added in version 2.87, May 1995.

## Title

If checked, enter any desired title. Maximum length is 78 characters. If blank, MsgBox95 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 MsgBox95Fun indicates which button was selected. In this example the "No" button was selected. See: [MsgBox95Fun](#), [MsgBox95Sub](#) for valid return codes.

## **Standard VB Message**

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

## Message95

Click the Message95 command to display the message created by it.

## Fonts

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

## Captions

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

## File

File	Sample Message	What's New..	Help	Test
	<u>M</u> essage Fonts...			Ctrl+O
	<u>C</u> ommand Captions...			Ctrl+C
	<u>S</u> elect an Alternate Icon...			Ctrl+S
	<u>S</u> elect 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 MsgBox3D Form			
	<u>R</u> egister with Ctl3DV2.DLL			
	Maximum Size is VGA Messages [640x480]			
✓	<u>N</u> o Message Separator <u>L</u> ine			
	Start MsgConvrt, the conversion utility..			
	<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 MsgBox95 Form](#)

[Register with Ctl3DV2.DLL](#)

[Maximum Size is VGA Message](#)

[No Message Separator Line](#)

[Start MsgConvrt...](#)



## **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 MsgWiz95.ini file, for example:

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

## **Save Selected Options in Ini File**

If checked, settings you selected while using MsgWiz95 are saved in the MsgWiz95.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 MsgWiz95. You could accomplish the same result by deleting the MsgWiz95.Ini file.**

## **Center Me**

Center this MsgWiz95 form.

## **Unload the MsgBox95 Form**

The MsgBox95 form is usually not unloaded after being displayed, it is hidden. Select this menu item to force an unload of the MsgBox95 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, MsgWiz95 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 MsgBox95 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], and 3D font effect.

In this example a 12 point, bold, inset light, Arial font is displayed.

## Sample Message

The message displayed by MsgBox95 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.

## 3D Font Effect

Click None, Raised, Inset, Light, or Heavy to demonstrate the 3D font effect. The menu item, 3D Font, can be used to accomplish the same results.

An inset, light font is shown in this example.

## Select a Different Font

Click this command button to display a font selection dialog screen. See: [Selecting a Font in the Demo](#) 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 MsgWiz95.

## Default Captions

These are the default command captions. They are used by MsgBox95 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.

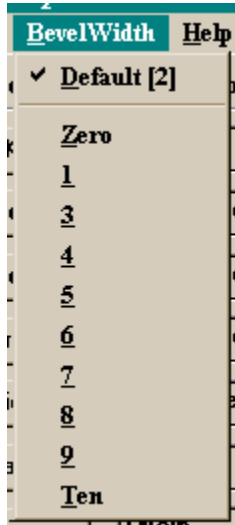
Three sets of sample captions are supplied with MsgWiz95. Click the Captions menu item for a list of sample replacement captions.

## 3D Font Effect

Click None, Raised, Inset, Light, or Heavy to demonstrate the 3D font effect. The menu item, 3D Font, can be used to accomplish the same results.

An inset, light font is shown in this example.

## Bevel Width



Select this menu item to change the command button's bevel width. Default is 2.

It is reported that a setting of 1 simulates the Windows 95 command button appearance.

See [Bevel Width](#) for details.

## **Use Default Settings**

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



## Captions

Captions	3D Font	BevelWidth
Factory Default		Ctrl+D
Some Samples		Ctrl+M
Spanish [Maybe]		Ctrl+S
French		Ctrl+F
✓ German		Ctrl+G

Four sets of sample command captions are supplied with MsgWiz95. Select one of the samples from this menu. Or select "Factory Default" to reset the commands to their "factory" setting.

Thanks go to Roger Doyon [CIS 74361,3542] who reviewed the French captions. The "updated" captions are included with MsgBox95 version 2.44 and later, first distributed on November 6, 1994.

And thanks to Holger Thornagel [CIS 100407,356] who supplied the German captions. They are provided with MsgBox95 version 2.72 and later, first distributed on March 12, 1995.

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 MsgBox95.

### Spanish Source:

Harper Collins Spanish Dictionary, 3rd Edition, 1992, 1993.  
ISBN 0-06-275504-8 (US), Library of Congress 91-36013.  
94/05/24 Millbrae City Library

### French Source:

Initial: Harrap's New Collegiate French and English Dictionary, 1982  
ISBN 0 8442 18596, Library of Congress 82-082075  
94/05/24 Millbrae City Library  
Updated by: Roger Doyon [CIS 74361,3542] 94/11/06

### German Source:

From: Holger Thornagel [CIS 100407,356] 95/03/12

## Font Name

Select a font name from the list of fonts. In this example 'MS Sans Serif' is highlighted.

## 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 MsgWiz95, MsgBox95 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\_FontNotBold As Integer  
Global MB3D\_FontItalic As Integer  
Global MB3D\_Font3D As Integer

#### **Command Fonts**

Global MB3D\_Commands\_FontName As String  
Global MB3D\_Commands\_FontSize As Single  
Global MB3D\_Commands\_FontNotBold As Integer  
Global MB3D\_Commands\_FontItalic As Integer  
Global MB3D\_Commands\_Font3D As Integer  
Global MB3D\_Commands\_No3D 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

#### **No Separator**

Global MB3D\_NoSeparator 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

#### **Register with Ctl3D**

Global MB3D\_UseCtl3D as Integer

#### **Command Button Bevel Width**

Global MB3D\_BevelWidth 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.

**MsgWiz95** will use **CTL3DV2.DLL**, if available, to display the browse common dialog screen with a 3D effect.

## **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 MsgBox95 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

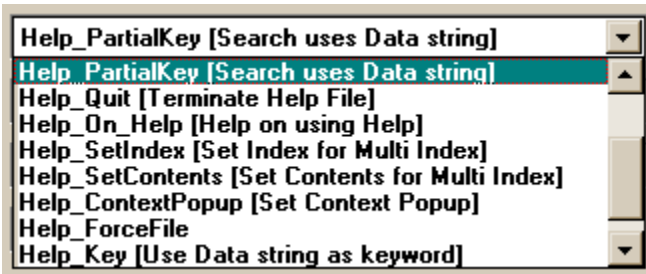
## Visual Basic Help Constants

The following is a list of Help Command constants defined in VB's Constant.Txt file:

```
Global Const HELP_CONTEXT = &H1
Global Const HELP_QUIT = &H2
Global Const HELP_INDEX = &H3
Global Const HELP_CONTENTS = &H3
Global Const HELP_HELPONHELP = &H4
Global Const HELP_SETINDEX = &H5
Global Const HELP_SETCONTENTS = &H5
Global Const HELP_CONTEXTPOPUP = &H8
Global Const HELP_FORCEFILE = &H9
Global Const HELP_KEY = &H101
Global Const HELP_COMMAND = &H102
Global Const HELP_PARTIALKEY = &H105
```

## 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.

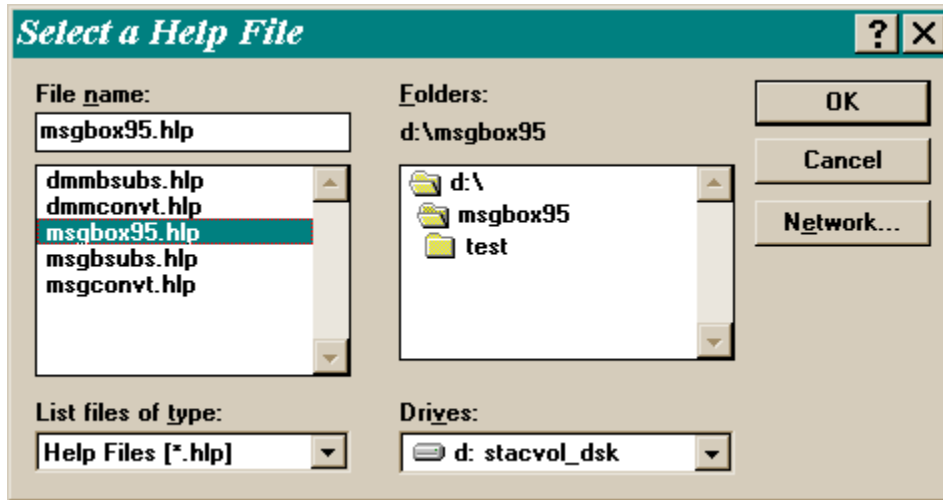
## Help File

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:



## 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, or 3D effect has been specified then this command is displayed with those command attributes. See: [Command Captions and Fonts](#) for details.

Click this command to "test" the help function. The same as pressing 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.



## MsgBox95 Source Components

### Used in Your Application:

MsgBox95.Frm	Message Form
MsgBox95.Frx	Binary Information, Mostly Icons
MsgBox95.Bas	Subroutines, Functions, Globals

### Used with MsgWiz95:

MsgBox95.Ico	Icon
MsgBox95.Hlp	This Help File
MsgGB95.Txt	Help File Constants
MsgWiz95.Frm	Generate VB Code and Demo Program
MsgWiz95.Frx	
MsgHlp35.Frm	Help Command Processor
MsgHlp95.Frx	
MsgFnt95.Frm	Message Fonts
MsgFnt95.Frx	
MsgCap95.Frm	Command Captions
MsgCap95.Frx	
MsgIco95.Frm	Alternate Icons
MsgIco95.Frx	
MsgMsg95.Frm	Edit the Message
MsgNew95.Frm	Display Whats New
MsgNew95.Frx	
MsgNew95.Bas	
MsgHst95.Bas	Development History
MsgWiz95.Bas	Subroutines
MsgIni95.Bas	Save Settings Subroutines
MsgWiz95.Mak	Make File
ReadMe.Txt	Current Information

The MsgBox conversion utility's components are described in the [MsgConv.Hlp](#) file.

