



MsgBox EASY

Just another little program that helps you a lot

Please choose topic:

[What is MsgBox EASY?](#)

[Layout and buttons](#)

[Configuration](#)

[How to use it?](#)

[Troubleshooting](#)

NOTE

Please note that if you use MsgBox EASY for your own fun or for developing freeware/PD it is free of charge.

If you use this program for developing commercial/shareware software, please send me \$10 US to support my work (I accept also useful registered/freeware Delphi 2.0 VCL components or fine registered shareware utils for Windows 95 you have written). I think this is fair. Thanks!

If you want to be informed when new version is released and also some other interesting apps, please send me mail and I'll add your name to my mailing list:

ahito@estpak.ee

Check out also my homepage where you can find info about me and download other stuff I've released:

<http://www.estpak.ee/~ahito>

Slander and money to:

Ahto Tanner
Karja 2-58
EE2820 Paide
ESTONIA

MsgBox EASY is Copyright 1996, Ahto Tanner. All rights reserved.



What is MsgBox EASY?

Just another little program that helps you a lot

This program is designed to help you quickly generate code for different message boxes and allows to save this code as template for future use. It has also many options, like use of your own variables, direct paste to VB project, use of automatic SELECT CASE block for responding user choices, use of VB 3, VB 4 constants or no constants at all, etc.

Also added 2 buttons for quick comment/uncomment of code blocks.

I tried to make the main form as small as possible so you can hold it always open. It suits to lower right corner of screen (below properties window), but anyone can place it anywhere he (she ;-) wants.

Program is released in 2 versions: one for 16-bit Windows 3.1 (**msg12-16.zip**) and other one for 32-bit Windows 95/NT (**msg12-32.zip**). If you need to develop for both Windows 3.1 & Windows 95, you can have both 2 MsgBox EASY executables in one directory, so you can share your templates between them. Both versions use private sections (accordingly Settings 16 and Settings 32) in single MSGBOX.INI, but the same Templates section for templates.

[How to use it?](#)

[Back to start](#)



Layout and buttons

Click on the item you would like to know about...



[Back to start](#)

[Configuration](#)

Generates messagebox code and copies it to the clipboard or pastes directly to VB project. Before copying/pasting you can test you dialog box by clicking on TEST button.



You can choose between copy/paste in [configuration](#) dialog box. On the Misc tab check/uncheck Direct paste to VB.... Note that the caption of this button is also changed.

This button is one of I have missed. It doesn't belong to msgbox function but is just a little bonus. It adds remark character before every row in block so you can change entire big or small block to comment with one mouse click. See also next Uncomment button.



If you comment/uncomment long text, please note, that it can take a little time while the block is processed, so please be patient.



Note that the comment symbol `%` MUST be the first in the line, otherwise Uncomment may not work correctly.

It removes all remark characters added by Comment button, from the block with just one mouse click.
See also previous Comment button.



If you comment/uncomment long text, please note, that it can take a little time while the block is processed, so please be patient.



Note also that it removes comment created with Comment button, not comments where the symbol is not the first symbol on line.

Click on the icon you want to be displayed in your dialog box. If you don't want any icon to be displayed and one button is pressed down, click on the pressed button again to switch it off.

Choose what buttons are displayed in your dialog box.

Set 2nd button in dialog to default. Default button is automatically chosen if user presses ENTER key or SPACE. This button has no effect if you have only one button in dialog.

Set 3rd button in dialog to default. Default button is automatically chosen if user presses ENTER key or SPACE. This button has no effect if you have only one or two buttons in dialog.

Tests the dialog with text, icon & caption you have coded so you can view if it looks right. If all is okay, press the COPY/PASTE button to copy/paste code for your dialog to clipboard/project.

Makes the dialog box system modal. This means that you must close the dialog before you can use any other programs. Use it only when critical error occurred that may impact other applications.

Makes the MsgBox EASY window to float above other windows (always on top - you know!)

Switches between 2 states of main window. Try out to see...

Copies the code header you defined in configuration panel to clipboard.

Opens the [configuration](#) panel.

Displays the name of currently active template. You can change it and save with new name. From combo box you can choose template you have already saved.

NOTE: When choosing new template from combo, changes made (if any) to the current template will be lost. You don't get warning... Before choosing next template, save current changes by pressing SAVE.

Here is the message you want to display in dialog box. Note that you can include optional variable %NL% followed by space for linefeed (Chr\$(10)). When copied to clipboard, this variable is replaced by variable you have chosen in configuration panel.



You can quickly insert %NL% variable in cursor position by pressing CTRL+ENTER.

Type here the text displayed in message box caption. If this row is blank, your application name is displayed instead.



You can enter one or more spaces here if you want your dialog without caption text.

Starts new template and sets all buttons to their defaults.

Deletes current template.

Saves current template

Saves current template with new name. Before choosing this button, you must open an existing template and change it's name to new name.



Configuration

Click on the item you would like to know about...

The screenshot shows a 'Configuration' dialog box with a blue title bar. It has four tabs: 'Misc.', 'Variables', 'Constants', and 'About'. The 'Misc.' tab is selected. Inside the 'Misc.' tab, there are several options: a checked checkbox for 'Play this sound file when action succeed:' with a text field containing 'copy.wav'; a text field for 'Add this line to the beginning of generated code:' containing 'Beep'; a checkbox for 'Minimize window after copy' which is unchecked; a checked checkbox for 'Generate SELECT CASE block'; a checked checkbox for 'Beep on dialog test'; a checked checkbox for 'Direct paste to VB instead of clipbrd'; a 'Header editor...' button; a 'TAB width:' label with a text field containing '4'; and a 'Default indent:' label with a text field containing '3'. At the bottom of the dialog are three buttons: 'Help', 'OK', and 'Cancel'.

Configuration

Misc. Variables Constants About

☒ Play this sound file when action succeed:
copy.wav

Add this line to the beginning of generated code:
Beep

☐ Minimize window after copy ☒ Generate Select CASE block

☒ Beep on dialog test ☒ Direct paste to VB instead of clipbrd

Header editor... TAB width: 4 Default indent: 3

Help OK Cancel

[Back to start](#)

Here you can write statement or sub to be called before message box is shown. This may be for example any sub that produces sound etc. If you don't want this kind of command, clear the text box.

```
Msg$ = Do you know me?
```

```
Capt$ = Hi
```

```
Beep
```

```
Resp% = MsgBox(Msg$, vbQuestion + vbYesNo, Capt$)
```

Check this if you want to hear something if you click Copy, Save or Copy code header... Then selected wave file will be played when action succeed.



You dont need to specify the full path to the file if the file is in the same folder as program itself.

When this is checked, then direct paste is enabled. This means that code is placed directly to your VB project instead of clipboard. Note that the code is placed with right indent level to your cursor (except VB 3, where the code is placed to the left margin - then use TAB with block to make it right level).

Check this if you want to hear standard windows beep when testing dialogues.

Here you can specify, whether to see SELECT CASE block after msgbox function or not.

```
Beep
```

```
Resp% = MsgBox(Do you know me?, vbQuestion + vbYesNo, Hi)
```

```
Select Case Resp%
```

```
    Case vbYes
```

```
        ` your code for YES button
```

```
    Case vbNo
```

```
        ` your code for NO button
```

```
End Select
```

Check this if you want each time after copy/paste operation automatically minimize MsgBox EASY window.

Tab level (indent) is number of tabs before the function begins:

```
Beep
```

```
Resp% = MsgBox(Do you know me?, vbQuestion + vbYesNo, Hi)
```

|-----|

Use it only when using VB 3.0 or VB 4.0 in non-paste mode. Then, when you paste your code from clipboard to project, it remains with defined indent level.



Its funny, but are you already know, that you can indent entire selected **block** of code at a time by pressing TAB? You can also de-indent block by pressing SHIFT-TAB.
Im writing it because I discovered this after a year of using VB... ;-)

This button shows the header editor window, where you can write your own standard header. This is useful if you want all your projects include similar style header with copyright, date, time, name etc.

Here is the tab stop width. You can configure this number in VB environment settings. Change it only when you are using VB 4 32-bit, otherwise it is changed automatically by program.

It is used to calculate right indent level if you paste directly to VB and want automatically right indent for pasted code.

Note that if you change your VBs tab stop width while MsgBox EASY is running, please restart MsgBox EASY so it can read new settings from VB.INI file. Do it only when you are NOT using VB 4 32-bit, otherwise change this number here to the same you chose in VB.



Configuration

Click on the item you would like to know about...

Configuration

Misc. **Variables** Constants About

☒ Use variables in generated code

Caption: Capt\$

Message: Msg\$

Response: Resp%

Variable for linefeed: vbCrLf

Help OK Cancel

[Back to start](#)

If checked, use of variables is enabled. Some programmers prefer this kind of variables. Note that you can use your own variable names.

With variables, code looks like this:

```
Msg$ = Do you know me?  
Capt$ = Hi  
Resp% = MsgBox(Msg$, 36, Capt$)
```

The same code without variables:

```
Resp% = MsgBox(Do you know me?, 36, Hi)
```

If variables are enabled, your var name for message box caption string:

```
Msg$ = Do you know me?
```

```
Capt$ = Hi
```

```
Resp% = MsgBox(Msg$, 36, Capt$)
```

If variables are enabled, your var name for message string:

```
Msg$ = Do you know me?
```

```
Capt$ = Hi
```

```
Resp% = MsgBox(Msg$, 36, Capt$)
```

If variables are enabled, your var name for response constant:

```
Msg$ = Do you know me?
```

```
Capt$ = Hi
```

```
Resp% = MsgBox(Msg$, 36, Capt$)
```

Variable for linefeed character. You can also specify Chr\$(10) if you don't want variable. This variable is used instead of %NL% if you used any %NL% in message text string.



Visual Basic 4 has built-in linefeed constant **vbLf** - use it!



Configuration

Click on the item you would like to know about...

The screenshot shows a 'Configuration' dialog box with a blue title bar. It has four tabs: 'Misc.', 'Variables', 'Constants' (which is selected and has a dashed border), and 'About'. The 'Constants' tab contains the following options:

- ☒ Use constants in generated code
 - ☐ Visual Basic 3.0 constants
 - ☒ Visual Basic 4.0 constants
- ☒ Constants in response SELECT CASE block

At the bottom of the dialog are three buttons: 'Help', 'OK', and 'Cancel'.

[Back to start](#)

If checked, use of constants is enabled. Some programmers prefer constants but in Visual Basic 4 all these constants are already globally declared so the use of them is very convenient.

With VB 4 constants, code looks like this:

```
Resp% = MsgBox(Do you know me?, vbQuestion + vbYesNo, Hi)
```

The same code without constants:

```
Resp% = MsgBox(Do you know me?, 36, Hi)
```

If the use of constants is enabled, you can choose to use VB 3 or VB 4 constants.

With VB 3 constants, code looks like this:

```
Resp% = MsgBox(Do you know me?, MB_ICONQUESTION + MB_YESNO, Hi)
```

With VB 4 constants, code looks like this:

```
Resp% = MsgBox(Do you know me?, vbQuestion + vbYesNo, Hi)
```

NOTE: VB 3 constants are not already declared so you must to declare them yourself.

If you choose from Misc. tab Response SELECT CASE block then you have possibility to have constants in that block.



How to use MsgBox EASY

Just another little program that is unbelievable simple to use

Wow! Its so simple to use, I decided not to write about this at all... (choose icon, choose buttons, choose default button, choose if modal, type message text, type caption, press copy/paste -- thats it!)



Actually there is one point you must remember. If you want to paste generated code directly to your VB project, this child window that gets the code must be last active window before you activate MsgBox EASY and press Paste button. For example if last active child window of your project was Properties, then MsgBox EASY tries to paste your code there. And dont forget to place your cursor exactly where you want the code to be pasted (you can choose even right indentation).



If something goes wrong, you can always select Undo from VB-s Edit menu to undo the last action.

[Back to start](#)

[Configuration](#)



Troubleshooting

Just another little program that doesnt trouble you a lot

Templates and other options are saved in file MSGBOX.INI in the same directory as the program. Is good idea to copy now and again this file to safe location, specially if you have large number of templates, because if something happens with original file you will lost your work (saved templates).

Program makes every time it exits copy of original file with extension BAK and if it detects at start-up corruption in original MSGBOX.INI file, it asks if you want to restore your settings from backup copy.

If you find a bug, please let me know, Im collecting them ;-)

[Back to start](#)

[Configuration](#)

