

# STBAR

## Visual Basic StatusBar Control (Version 2.1)

### 1. Overview

StBar is a custom control, which allows to add a status line to your Visual Basic applications. The main idea behind this control is to make an extra instance of the control for any section (information field , NUM and CAPS indicators, ...) of the status line. As an extra bonus you get improved menu and task switch processing.

### 2. Installation

To use the control, simply copy STBAR.VBX to the windows system directory or to another directory, which is in specified in your PATH environment variable. You may also want to customize the STBAR.INI file (see below) and copy it to your windows directory.

### 3. How to use the control

To add a status line to an application:

add STBAR.VBX to the project (File/Add File menu).

for each section, add an instance of the control to the form (double click on the toolbar bitmap or click and draw as for any other control). The status bar sections are automatically placed at the bottom of the form (don't draw status bar controls within a container control). Sizing the form also stretches or shrinks the status line (in fact, only the width of the leftmost StBar instance changes). This automatic placing and sizing works in design and run mode.

set the properties for each status line section (see below).

### 4. StBar Properties

#### 4.1 Standard Properties

The following properties work as for any other Visual Basic control and need no further explanation:

- CtlName
- Index
- Visible (see StBarVisible function described below)
- Parent
- Tag
- Caption (This is the 'contents' of that status line section, automatic for 'indicators')
- Font...

#### 4.2 Automatic Properties

The following properties are set automatically (changes are also automatic in response to resizing the form or modifying system colors):

- BackColor
- Left, Top, Width, Height

#### 4.3 Special Properties

These are the properties specific to StBar:

- DrawMode:

gives a 3-D or flat look to the corresponding status line section.

- Indicator:

makes a section of a status bar an indicator for the NUM, CAPS, SCROLL or INSERT key (insert is not 'hardware' but 'form' dependent). The actual abbreviation for each indicator can be modified by changing STBAR.INI (don't use more than 4 characters). If you don't supply definitions in STBAR.INI, default names are used).

- LinkedTo (Syntax: FormName.ControlName)

links that section to another StBar instance on another form. If you only want to have a (visible) status bar on one (main) form of your application, set the visible property of the status bar sections on other forms to FALSE and link them to corresponding sections on the main form.

You can get the same result e.g. with the following code

```
fmMain.StBar1.Caption = "Status Bar Text"
```

but we also have an extended text box control (XEDIT.VBX) and clones of VB's option button and check box controls (CHOICE.VBX). These controls have a special 'StBarText' property (and also optimize the handling of the insert key), and for them the LinkedTo property works very well.

## 5. Events

### 5.1 Events fired only to the first section of a status line

...\_AppActivate(Activate):

reflects the WM\_APPACTIVATE message for the status bars form.

...\_InitMenu():

menu processing has been activated for the form (WM\_INITMENU); this is the perfect place to do menu 'manipulations'.

...\_MenuSelect (Closing As Integer, MenuCaption As String):

a menu entry has been selected/closed (closing <> 0); use the menu caption to identify the menu and to put an explanation in the status bar. This works also for the system menu (with special MenuCaptions !!!).

...\_EmptyCaption

This event is triggered, if the status bar caption gets empty. This is usefull, if the status bar is manipulated by other controls (e.g. xEdit, xOption, xCheck, ...), and the programmer wants to restore status information, when it is cleared.

### The following event is supported only for VB2

...\_SelectedTag (hWndForm As Integer, MenuTag As String):

a menu item has been selected; use the tag property of the menu to identify the menu and to put an explanation in the status bar (this may be the tag). This works also for the system menu (s. a.). The window handle ist that of the window, to which the messages is directed. This is useful for MDI application to differentiate e.g between the system menu's Close item for the application window (Quit application) and for child windows (Close document). Not that this works also for maximized MDI child, though Windows does not send the WM\_MENUSELECT message to the child (in fact, the child's system menu has changed to an entry in the main window's menu).

## 5.2 Events fired to indicator sections

...\_SetIndicator (State As Integer):

reflects the changing of the indicator state (ON/OFF)

## 5.3 special MenuCaption/MenuTag strings for the MenuSelect and SelectedTag events:

SYS\_POPUP: the system menu  
SYS\_SIZE: the SIZE menu item  
SYS\_MOVE: the MOVE menu item  
SYS\_MINIMIZE: the MINIMIZE menu item  
SYS\_MAXIMIZE: the MAXIMIZE menu item  
SYS\_CLOSE: the CLOSE menu item (s.a. for MDI applications)  
SYS\_RESTORE: the RESTORE menu item  
SYS\_NEXT: the NEXT WINDOW menu item (MDI child)  
SYS\_TASKLIST: the TASKLIST menu item

there is one other special value for the SelectedTag event:

SYS\_SWITCH: menu items in the window list of MDI applications

Please note, that SYS\_RESTORE is also sent, if the restore button in the application menu is selected for a maximized MDI child (we are very proud of this !!!).

## 6. Functions

There is one exported function in the DLL, StBarVisible, which used to show or hide the complete status line. Look at the sample for its declaration and on how to use it.

## 7. Extensions for Visual Basic 2.0

Properties:

StBar controls have the hWnd property.

StBar controls have the ValueProp property (caption), i. e. you can write sbMain = "Hello" instead of sbMain.caption = "Hello".

MDI forms:

With MDI the user expects to have a status line at the bottom of the main window. How can you achieve this?

draw a picture control on the MDI form (align to bottom, no border). Make it high enough to draw the StBar instances INSIDE THE PICTURE control.

The MDIForm\_Load procedure should contain something like this:

```
picStBar.Height = sbMain.Height
```

The rest is automatic !!! Try it out by adding a status bar to the MDI sample of VB2.

**Starting with version 2.1, the MenuSelect event and the SelectedTag event are also supported for the system menu of child windows (s. a.).**

## 9. Licence agreement

You may distribute free of charge any application developed using STBAR.VBX. But don't give away the 'full' version of StBar. Instead, put the runtime version STBARD.VBX (renamed to STBAR.VBX) on your distribution diskettes.

I hope, you will enjoy this version of STBAR.VBX. Please send any comments to:

Bernd Beekes  
Eugen Haas GmbH & Co. KG  
Gummersbacher Str. 44-48  
D 5270 Gummersbach

CompuServe ID: 100031,2063

Phone: ..49 2261 300185

..49 2261 55369

Fax: ..49 2261 300168

Other controls available:

xEdit: enhanced TextBox, centralized field validation and message control

Choice: OptionButton/CheckBox with status bar property.

If you are building (large) database applications, ask for VB/Express, our complete development environment for Visual Basic, including VSQL (virtual SQL interface) and DLGBLD.VBX, the dialog builder custom control.