# DBPush v1.9 March 1995

**Description**    The DBPush command button control emulates the 3D command button control (SSCommand) supplied with Visual Basic Professional Edition 3.0. In addition it supports several enhancements to SSCommand.

**File Name**    DBPUSH.VBX

**Object Type**    DBPush

**Remarks**    This document describes only the properties where DBPush differs from SSCommand.

**About**    This control was developed by Dan Byström. For more information, contact me at e-mail: "dan.bystrom@adb-partner.it-invest.se" or phone: +46 708 68 65 78 (no support calls, please). I would be happy to discuss development of customised VBX'es or OCX'es for you.

**Distribution**    You have the right to do whatever you want with DBPush, *as long as you don't attempt to modify any of its code*. "Do whatever you want" includes using DBPush in your own commercial applications and distributing it for free.

When the control is loaded in design mode a message is sometimes displayed. *This message may not be removed or changed in any way*. Anyway, the message won't ever appear at run-time.

I'm giving DBPush away for free. If you should decide to use it in an application of yours, this means that I have saved you a whole lot of trouble, time and $$$ doing it yourself. Therefore I think a nice gesture would be to include some sort of credit in your application's about box and/or documentation. You could include the name of the VBX, my name and my e-mail address. I would be glad if you did this. Anyway - happy VB programming with DBPush!

**Revision history**
Oct 94:  Beta release.
    Nov 94:  BUGFIX: Right mouse button no longer generates a Click event.
        Library name changed from PUSH to DBPUSH.
        Version info is now included in the VBX.
        New property added: *ForceDown*.
        New property added: *CornerColor*.
    v1.1:  New property added: *ForceUp*.
        New property added: *WasDblClick*.
        New property added: *VertAlignment*.
    v1.2:  A minor error in the default setting of *VertAlignment* was fixed.
        *Placement* may now be "Text on top of picture" as well.
    v1.3:  New property added: *PictureDisabled*.
        New property added: *ForeColorDisabled*.
        New property added: *Font3DDisabled*.
        Added exported function for easy version control.
    v1.4:  New property added: *Style*.
        New property added: *CustomCursor*.
    v1.5:  BUGFIX: Two pictures (if used) were not released from memory.
        New property added: *FocusRect*.
        New property added: *OutlineSize*.
        The property *Outline* is superseded and doesn't show up in design mode.
        New property added: *Transparent*.
        New property added: *TransColor*.

The button is now first drawn in an off-screen buffer to prevent flicker.

Double pipe characters may now be used instead of CRLF in caption.

v1.6:  BUGFIX: 256-color pictures were handled incorrectly in v1.5.
New property added: *PictureFromEdge*.
New property added: *PredefPicture*.
Added new *Style*: elliptical buttons.

v1.7:  Some minor glitches were fixed.
An about box was finally added.

v1.8:  On-line help was added, thanks to David A. Karp.
New property added: *AutoPictureDisabled*.

v1.9:  Added four more styles for *AutoPictureDisabled*.
Added two more predefined pictures.

## Properties

| | | | |
|---|---|---|---|
| *Alignment | *AutoPictureDisabled | *BackColor | *BackColorDown |
| *BevelWidth | *Caption | *CornerColor | *CustomCursor |
| DragIcon | DragMode | Enabled | *FocusRect |
| *Font3D | *Font3DDisabled | FontBold | FontItalic |
| FontName | FontSize | FontStrikethru | FontUnderline |
| *ForceDown | *ForceUp | ForeColor | *ForeColorDisabled |
| Height | hWnd | Index | Left |
| MousePointer | Name | *OutlineSize | Parent |
| *Picture | *PictureDisabled | *PictureDown | |
| *PictureFromEdge | | | |
| *Placement | *PredefPicture | *Style | TabIndex |
| TabStop | Tag | *Top | *TransColor |
| *Transparent | *VertAlignment | Visible | *WasDblClick |
| Width | | | |

\* = The property applies only to DBPush (or offers some improvement, like the *Caption* property).

## Alignment Property

**Description**    Determines the alignment of the caption text within the push button.

**Remarks**    The *Alignment* property settings are:

| Setting | Description |
|---|---|
| 0 | The caption text appears left-justified. |
| 1 | The caption text appears right-justified. |
| 2 | (Default) The caption text appears (horizontally) centered. |

**Data Type**    Integer (enumerated)

## AutoPictureDisabled Property

**Description**    Attempts to automatically give the picture (if any) a "disabled" look when the *Enabled* property is set to *False*.

**Remarks**     The *AutoPictureDisabled* property settings are:

| Setting | Description |
| --- | --- |
| 0 | (Default) No attempt is made to give any picture a disabled look. |
| 1 | The picture's outer contour appears inset on the screen. |
| 2 | The picture's outer contour appears raised off the screen. |
| 3 | The picture itself appears inset on the screen. |
| 4 | The picture itself appears raised the screen. |
| 5 | All contours appears inset on the screen. Some contours will be two pixels thick. |
| 6 | All contours appears raised off the screen. Some contours will be two pixels thick. |
| 7 | All contours appears inset on the screen. Some contours will disappear. |
| 8 | All contours appears raised off the screen. Some contours will disappear. |

The first four styles (operating on the whole picture) requires the *TransColor* property to be set to the picture's background color, even if the transparent feature isn't used (this isn't necessary for icons, however).

**Data Type**     Integer (enumerated)


# BackColor Property

**Description**     Determines the background color of the button.

**Remarks**     This property is set to light gray by default and it probably looks best that way.

**Data Type**     Long


# BackColorDown Property

**Description**     Determines the background color of the button when the button is depressed.

**Remarks**     This property is set to light gray by default and it probably looks best that way, unless you want to use the button as a two-state button.

**Data Type**     Long


# BevelWidth Property

**Description**     Sets or returns the width of the bevel along the four sides of the command button to determine the height of the three-dimensional shadow effect.

**Remarks**     The setting for this property determines the number of pixels used to draw the bevel that surrounds the command button.

**Data Type**     Integer

# Caption Property

**Description**     The button's caption text.

**Remarks**         Unlike SSCommand, DBPush is able to display several lines of caption text. If the caption text doesn't fit on one line (or if it contains CRLF breaks) it is automatically wrapped onto several lines. Beginning with DBPush v1.5, a pair of pipe characters (or "vertical bars") may be used instead of CRLF. This makes it possible to create hard line breaks in design mode.

**Data Type**       String

# CornerColor Property

**Description**     Determines the color of the corners on the outline. Instead of implementing a *RoundedCorners* property like SSCommand (which always uses light gray to draw smooth corners), DBPush lets you decide what color to use.

**Remarks**         This property is set to black by default which means that the control appears to have sharp edges. To mimic a *RoundedCorners* property, set *CornerColor* to the same value as the parent form's background color.

**Data Type**       Long

# CustomCursor Property

**Description**     Sets a custom cursor shape to use when the mouse is moved over the button. *CustomCursor* takes precedence over the *MousePointer* property.

**Remarks**         Just like the *DragIcon* property, the cursors must be saved as icons in an .ICO file. This is because VB handles .ICO files by default, but not .CUR files. This property is somewhat more restrictive than *DragIcon*; <u>only</u> the white color in the icon translates to white color in the cursor - all other colors translate to black.

**Data Type**       Picture (Icon only)

# DBPushVersion Function

**Description**     This <u>function</u> may be used to investigate the version of the system's DBPUSH.VBX and supplying the user with an appropriate error message if an old version is found.

**Remarks**         The major version number is returned in the high byte and the minor version number is returned in the low byte. For an example, refer to the last section of this document.

**Data Type**       Integer

# Enabled Property

**Description**     Determines whether the control is enabled or not.

**Remarks**    The caption text is *not* grayed like normal buttons when the control is disabled. Instead, the properties *Font3DDisabled*, *ForeColorDisabled* and *PictureDisabled* may assist you in presenting an alternate disabled look.

**Data Type**    Integer (Boolean)

# Font3D Property

**Description**    Sets or returns the three-dimensional style of the command button caption..

**Remarks**    The following table lists the *Font3D* property settings for the 3D command button control:

| Setting | Description |
| --- | --- |
| 0 | (Delault) No shading. Caption is displayed flat (not three-dimensional). |
| 1 | Raised with light shading. Caption appears raised off the screen. |
| 2 | Raised with heavy shading. Caption appears more raised. |
| 3 | Inset with light shading. Caption appears inset on the screen. |
| 4 | Inset with heavy shading. Caption appears more inset. |

**Data Type**    Integer (enumerated)

# Font3DDisabled Property

**Description**    Determines the 3D style of the caption when the button is in a disabled state.

**Remarks**    Try to set *ForeColorDisabled* to dark gray and *Font3DDisabled* to "3 - Inset with light shading".

**Data Type**    Integer (enumerated)

# FocusRect Property

**Description**    Determines whether a focus rectangle shall be drawn when the button gets focus.

**Remarks**    This property is True by default.

**Data Type**    Integer (Boolean)

# ForceDown Property

**Description**    Forces the button to be drawn in a depressed state.

**Remarks**    This property may be used to mimic the behaviour of a check button. Just toggle the property value (DBPush1.ForceDown = Not DBPush1.ForceDown) in the Click event.

**Data Type**    Integer (Boolean)

## ForceUp Property

**Description**     Forces the button to always be drawn in up state.

**Remarks**     By enabling this property the button no longer behaves like a command button, but more like a specialised picture control. When this property is set, no focus rectangle is drawn around the picture or the caption.

This property does not affect the events the button receives in any way.

**Data Type**     Integer (Boolean)


## ForeColorDisabled Property

**Description**     Determines the caption color when the button is in a disabled state.

**Remarks**     Try to set *ForeColor* to dark gray and *Font3DDisabled* to "3 - Inset with light shading".

**Data Type**     Long


## OutlineSize Property

**Description**     Determines the size of the black outline (or border) around the button. Set this property to zero to remove the outline.

**Remarks**     This property may be used to indicate a default button (by setting *OutlineSize* to two).

**Data Type**      Integer


## Picture Property

**Description**     Specifies a bitmap or an icon to display on the command button.

**Remarks**     The *Placement* property determines where the picture will be placed with respect to the caption.

**Data Type**     Picture


## PictureDisabled Property

**Description**     Determines the picture to be displayed when the button is disabled.

**Remarks**     If no picture is assigned to this property, the *Picture* property is used for the disabled button as well.

**Data Type**     Picture

# PictureDown Property

**Description**    Determines the picture to be displayed when the button is depressed.

**Remarks**    If no picture is assigned to this property, the *Picture* property is used for the depressed button as well.

**Data Type**    Picture


# PictureFromEdge Property

**Description**    Determines the distance between the button's edge and the picture given in pixels.

**Remarks**    The distance may be a negative number.

**Data Type**    Integer


# Placement Property

**Description**    Determines the placement of the picture with respect to the caption text.

**Remarks**    The *Placement* property settings are:

| Setting | Description |
| --- | --- |
| 0 | (Default) The picture appears horizontally centred at the top of the button and the caption text appears below. |
| 1 | The picture appears vertically centred at the right of the button and the caption text appears to the left. |
| 2 | The picture appears horizontally centred at the bottom of the button and the caption text appears above. |
| 3 | The picture appears vertically centred at the left of the button and the caption text to the right. |
| 4 | The picture appears both vertically and horizontally centred and the caption will be drawn on top of the picture if necessary. |

**Data Type**    Integer (enumerated)


# PredefPicture Property

**Description**    Selects one of three predefined pictures.

**Remarks**    The *PredefPicture* property settings are:

| Setting | Description |
| --- | --- |
| 0 | (Default) No predefined picture. |
| 1 | OK. |
| 2 | Cancel. |
| 3 | Help. |
| 4 | Yes. |

| 5 | No. |
|---|---|

Each setting has three associated pictures: up, down and disabled. Any picture specified by *Picture*, *PictureDown* or *PictureDisabled* takes precedence over *PredefPicture*. The "down" pictures are *only* used if *BevelWidth* is two or less, otherwise the "up" picture is used!

**Data Type**  Integer (enumerated)

## Style Property

**Description**  Gives the button a different style than normal command buttons.

**Remarks**  The *Style* property settings are:

| Setting | Description |
|---|---|
| 0 | (Default) Normal. |
| 1 | Sink deep. When the button is depressed it appears to sink below the surface of the form. This works best if *OutlineSize* is zero. |
| 2 | Elliptical button. (The *CornerColor* property must be set to the form's background color.) |
| 3 | Elliptical button with "sink deep" style. |

**Data Type**  Integer (enumerated)

## TransColor Property

**Description**  This property is used to make one color of a picture transparent, which means that this particular color is replaced with the button's background color.

**Remarks**  For this property to have any effect, the *Transparent* property must be enabled. Instead of using this feature you may consider repainting the picture, since both extra execution time and system resources are needed each time the button is repainted. This property is however useful when the button uses a different background color when in a depressed state (using the *BackColorDown* property).

**Data Type**  Long

## Transparent Property

**Description**  This property is used to enable the *TransColor* property.

**Remarks**  See the *TransColor* property.

**Data Type**  Integer (Boolean)

## VertAlignment Property

**Description**  Aligns the caption text vertically within the area not occupied by a picture.

**Remarks**      The *VertAlignment* property settings are:

| Setting | Description |
|---------|-------------|
| 0 | Caption is aligned at the top. |
| 1 | Caption is aligned at the bottom. |
| 2 | Caption is centred vertically. |

Every time the *Picture* property or the *Placement* property is altered, this property changes to the (static) alignment used before version 1.1. This is done purely to ensure version compatibility. Beware!

**Data Type**      Integer (enumerated)


## WasDblClick Property

**Description**      In the Click event, this property may be used to determine if the Click originated from a DblClick.

**Remarks**      Command buttons should normally not respond to DblClicks, but this is nevertheless a method to do it.

**Data Type**      Integer (Boolean)


## Group push buttons

It is possible to use DBPush as a group push button. If you want only one button in a group to be down at the same time, just make a control array out of the buttons of a group and include the following code in the click event:

```
Sub DBPush1_Click (Index As Integer)
 DBPush1(nCurrentButtonDown).ForceDown = False
 DBPush1(Index).ForceDown = True
 nCurrentButtonDown = Index
End Sub
```

Also declare this somewhere (in the form or a module):

```
Dim nCurrentButtonDown As Integer
```

You may also want to set the *BackColorDown* property to &H00E0E0E0& in buttons used as group push buttons.


## Palette considerations

DBPush is not made palette aware. This is done on purpose; the user would probably not like to see palette shifts each time the focus is moved to a new button. This doesn't mean that you can't use 256 color pictures, however. The only thing you need to do is to have the same palette as the button uses realised in some other way. The easiest way to do this is to put a 1x1 picture with the same palette in an image control on the form and move it outside the visible area.

## Cancel and Default buttons

The standard VB command button have the *Cancel* and *Default* properties which can't(?) be mimicked by a custom control. A primitive work-around is to place normal command buttons outside the visible area of the form, set their *Cancel* and *Default* properties respectively and let their click events call the corresponding DBPush buttons. Another way is to check for Enter and Esc in the Form's KeyPress event.

## Version control

The following code shows an easy way to check the version of DBPUSH.VBX before its properties are accessed by VB. In a global module, put the following declaration:

```
Declare Function DBPushVersion Lib "dbpush.vbx" () As Integer
```

Then use a  Sub Main() as your program's entry point:

```
Sub Main()
 If Hex$(DBPushVersion()) < "0190" Then
   MsgBox "Your DBPUSH.VBX is too old for this application!", 16
   End
 End If
'Load your main form here
End Sub
```