

What's the BS_PUSHLIKE button style for?



Raymond Chen

One of the lesser-known button styles is `BS_PUSHLIKE`.

Makes a button (such as a check box, three-state check box, or radio button) look and act like a push button. The button looks raised when it isn't pushed or checked, and sunken when it is pushed or checked.

In other words, you add this style to a check box or radio button to make it *look like* a push button even though it will continue to *act like* a check box or radio button.

	Check box	Check box push-like	Radio button	Radio button push-like
Unselected	Label	Label	Label	Label
Selected	Label	Label	Label	Label

Aside from the appearance, the other checkbox and radio button behaviors are preserved. Each time you click an automatic check box, it toggles between unselected and selected; it's just that instead of hiding and showing the check-mark, it pops the button out and pushes the button in. Similarly, the automatic radio button becomes selected when you click on it and becomes deselected when you select another radio button in the group. All that changed is the visuals.

Let's illustrate this with a quick sample program. First, we'll use traditional check boxes and radio buttons.

```

1 DIALOG 64, 64, 100, 70
STYLE WS_CAPTION | WS_SYSMENU
CAPTION "Demo"
FONT 8, "MS Shell Dlg"
BEGIN
    AUTORADIOBUTTON "Search &forward", 100, 4, 9,
        75, 14, WS_GROUP | WS_TABSTOP
    AUTORADIOBUTTON "Search &backward", 101, 4, 27,
        75, 14
    AUTOCHECKBOX "&Ignore case", 102, 4, 45,
        75, 14, WS_TABSTOP
END
INT_PTR CALLBACK DlgProc(
    HWND hDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch (uMsg) {
        case WM_INITDIALOG: return TRUE;
        case WM_CLOSE: EndDialog(hDlg, 1); break;
    }
    return FALSE;
}
int WINAPI WinMain(HINSTANCE hinst, HINSTANCE hinstPrev,
    LPSTR lpCmdLine, int nShowCmd)
{
    DialogBox(hinst, MAKEINTRESOURCE(1), 0, DlgProc);
    return 0;
}

```

When you run this program, you get two radio buttons and a check box, and there's nothing special about them at all. But add the `BS_PUSHLIKE` style...

```

1 DIALOG 64, 64, 100, 70
STYLE WS_CAPTION | WS_SYSMENU
CAPTION "Demo"
FONT 8, "MS Shell Dlg"
BEGIN
    AUTORADIOBUTTON "Search &forward", 100, 4, 9,
        75, 14, WS_GROUP | WS_TABSTOP | BS_PUSHLIKE
    AUTORADIOBUTTON "Search &backward", 101, 4, 27,
        75, 14, BS_PUSHLIKE
    AUTOCHECKBOX "&Ignore case", 102, 4, 45,
        75, 14, WS_TABSTOP | BS_PUSHLIKE
END

```

and now the radio buttons and check box look like push buttons. But if you click on them, they still behave like two radio buttons and a check box. if you select "Search forward", then "Search backward" automatically de-selects itself, and vice versa. Each time you click on "Ignore case", it toggles between pushed-in and popped-out.

The visuals are kind of confusing, however, if you have enabled Windows XP visual styles, for when you hover over a button, the theme engine will draw the button in its “hover” appearance, which causes it to pop out even though the button really is pushed in. You have to move the mouse away from the button to see that it really is pushed in. Personally, I consider this a bug. The situation in Windows Vista is only slightly better; it’s still pretty bad.

Push-like buttons are not used much since they duplicate the semantics of radio buttons and check boxes while presenting a deceptive visual appearance. Since they look like push buttons, users naturally expect them to behave like push buttons, and then when they don’t, things get confusing. The only case I can think of off the top of my head where push-like buttons actually match with user expectations is in toolbars, where there is a long-standing convention of using push-like buttons in toolbars instead of radio buttons and check boxes. (Fortunately, toolbars manage their own buttons and don’t suffer from the confusing hover-appearance behavior I discussed above.)

Raymond Chen

Follow

