

## TrackPopupMenu Function

### TrackPopupMenu Function

The TrackPopupMenu function displays a shortcut menu at the specified location and tracks the selection of items on the menu. The shortcut menu can appear anywhere on the screen.

#### Syntax

```
BOOL TrackPopupMenu(
    HMENU hMenu,
    UINT uFlags,
    int x,
    int y,
    int nReserved,
    HWND hWnd,
    CONST RECT * prcRect
);
```

Adding items to Windows Explorer context (right-click) menu easily &ndash; How to add them ?

Add items to Explorer Shell context menu easily with Windows Explorer Context Menu. This powerful .Net component for custom items adding to Windows Explorer context menu will add all your application entries to Windows Explorer context menu. This .Net component and Visual Basic .NET support include detailed C# / VB.NET samples, tutorials , user-friendly manuals and support all you may need to add your entries to Explorer context menu :

- Add all your items to Windows Explorer Shell context menu to be shown on any Windows operating system (all OS are supported &ndash; Windows XP, Vista, x64 , etc.)
- Add all your items to Windows Explorer context menu to be shown in any way - with custom caption and your custom icon, as separator or sub-menu
- Add items to Windows Explorer context menu to be shown for all files or shown only for computer files of particular type (for example, only for .DOC , .MP3,.WMA,.AAC , .AVI media files)
- Add your program entries to Windows Explorer Shell context menu, sub-menus, sub-menus of unlimited depth and add to Explorer context menu entries of all types

Explorer Shell Context Menu - is a .Net framework component that support all you may need to insert your application items to Windows Explorer Shell context menu - in a fast and a very easy way. Add your program items to Explorer Shell context menu right now &ndash; add items to context menu fast , easy and exactly as you prefer :

Add items to Windows Explorer context menu easily &ndash; How it may be done ?

Add items to Windows Explorer Shell context menu easily with Windows Explorer Context Menu. This powerful .Net component for your own, custom items adding to Explorer context menu will add all your custom application entries to the Windows Explorer context menu. It , C++ and VB.NET support include detailed C# and VB.NET samples, tutorials and support all you may need to add your items to context menu :

- Add all your items to Windows Explorer context menu to be shown on any Windows operating system (all OS are

supported &ndash; XP, Vista, Windows x64 , etc.)

- Add any type of items to Windows Explorer Shell context menu to be shown in any way - with your custom caption and your custom icon, as separator or sub-menu
- Add items of any types to Windows Explorer Shell context menu to be shown for all files or shown only for computer files of particular type (for example, only for .DOC , .MP3,.WMA,.AAC , .AVI files)
- Add your program entries to Explorer context menu, sub-menus, sub-menus of unlimited depth and add to Explorer context menu entries of all types

Windows Explorer Shell Context Menu - is a .Net component that support all you may need to insert all your application items to the Explorer Shell context menu - in a fast and easy way. Add all your items to Windows Explorer Shell context menu right now &ndash; add entries to context menu fast and exactly as you want :

## TrackPopupMenu - Introduction

TrackPopupMenu is a function used in old Windows operating systems to add items to Explorer context menu. Because this method can add item to Explorer context menu only for Windows 95 / Windows 98 (not on XP, Vista, x64 - 64-bit Windows), to add items to Windows Explorer Shell context menu you should use, according to Microsoft guidelines, appropriate .Net component - Windows Explorer Context Menu. Explorer Context Menu will add item of any type (add item with text, add item with text and icon, add item with submenu and add separator) to Windows Explorer context menu in a very easy way.

## Parameters

### hMenu

[in] Handle to the shortcut menu to be displayed. The handle can be obtained by calling CreatePopupMenu to create a new shortcut menu, or by calling GetSubMenu to retrieve a handle to a submenu associated with an existing menu item.

### uFlags

[in] Use zero or more of these flags to specify function options.

Use one of the following flags to specify how the function positions the shortcut menu horizontally.

#### TPM\_CENTERALIGN

If this flag is set, the function centers the shortcut menu horizontally relative to the coordinate specified by the x parameter.

#### TPM\_LEFTALIGN

If this flag is set, the function positions the shortcut menu so that its left side is aligned with the coordinate specified by the x parameter.

#### TPM\_RIGHTALIGN

Positions the shortcut menu so that its right side is aligned with the coordinate specified by the x parameter.

Use one of the following flags to specify how the function positions the shortcut menu vertically.

#### TPM\_BOTTOMALIGN

If this flag is set, the function positions the shortcut menu so that its bottom side is aligned with the coordinate specified by the y parameter.

#### TPM\_TOPALIGN

If this flag is set, the function positions the shortcut menu so that its top side is aligned with the coordinate specified by the y parameter.

#### TPM\_VCENTERALIGN

If this flag is set, the function centers the shortcut menu vertically relative to the coordinate specified by the y parameter.

Use the following flags to determine the user selection without having to set up a parent window for the menu.

#### TPM\_NONOTIFY

If this flag is set, the function does not send notification messages when the user clicks on a menu item.

#### TPM\_RETURNCMD

If this flag is set, the function returns the menu item identifier of the user's selection in the return value.

Use one of the following flags to specify which mouse button the shortcut menu tracks.

**TPM\_LEFTBUTTON**

If this flag is set, the user can select menu items with only the left mouse button.

**TPM\_RIGHTBUTTON**

If this flag is set, the user can select menu items with both the left and right mouse buttons.

Windows 98/Me: Use any reasonable combination of the following flags to modify the animation of a menu. For example, by selecting a horizontal and a vertical flag you can achieve diagonal animation.

**TPM\_HORNEGANIMATION**

Animates the menu from right to left.

**TPM\_HORPOSANIMATION**

Animates the menu from left to right.

**TPM\_NOANIMATION**

Displays menu without animation.

**TPM\_VERNEGANIMATION**

Animates the menu from bottom to top.

**TPM\_VERPOSANIMATION**

Animates the menu from top to bottom.

For any animation to occur, the SystemParametersInfo function must set SPI\_SETMENUANIMATION. Also, all the TPM\_\*ANIMATION flags, except TPM\_NOANIMATION, are ignored if menu fade animation is on, See the SPI\_GETMENUFADE flag in SystemParametersInfo.

Windows 98/Me, Windows 2000/XP, Vista: Use the TPM\_RECURSE flag to display a menu when another menu is already displayed. This is intended to support context menus within a menu.

Windows XP, Vista: To have text layout from right-to-left, use TPM\_LAYOUTRTL. By default, the text layout is left-to-right.

x

[in] Specifies the horizontal location of the shortcut menu, in screen coordinates.

y

[in] Specifies the vertical location of the shortcut menu, in screen coordinates.

nReserved

Reserved; must be zero.

hWnd

[in] Handle to the window that owns the shortcut menu. This window receives all messages from the menu. The window does not receive a WM\_COMMAND message from the menu until the function returns. If you specify TPM\_NONOTIFY in the uFlags parameter, the function does not send messages to the window identified by hWnd. However, you must still pass a window handle in hWnd. It can be any window handle from your application.

prcRect

Ignored.

**Return Value**

If you specify TPM\_RETURNCMD in the uFlags parameter, the return value is the menu-item identifier of the item that the user selected. If the user cancels the menu without making a selection, or if an error occurs, then the return value is zero.

If you do not specify TPM\_RETURNCMD in the uFlags parameter, the return value is nonzero if the function succeeds and zero if it fails. To get extended error information, call GetLastError.

**Remarks**

Note Call GetSystemMetrics(SM\_MENUDROPALIGNMENT) to determine the correct horizontal alignment flag (TPM\_LEFTALIGN or TPM\_RIGHTALIGN) and/or horizontal animation direction flag (TPM\_HORPOSANIMATION or TPM\_HORNEGANIMATION) to pass to TrackPopupMenu or TrackPopupMenuEx. This is essential for creating an optimal user experience, especially when developing Microsoft Tablet PC applications.

To specify an area of the screen the menu should not overlap, use the TrackPopupMenuEx function

To display a context menu for a notification icon, the current window must be the foreground window before the application calls TrackPopupMenu or TrackPopupMenuEx. Otherwise, the menu will not disappear when the user clicks

outside of the menu or the window that created the menu (if it is visible). However, when the current window is the foreground window, the second time this menu is displayed, it displays and then immediately disappears. To correct this, you must force a task switch to the application that called TrackPopupMenu. This is done by posting a benign message to the window or thread, as shown in the following code sample:

```
SetForegroundWindow(hDlg);

// Display the menu
TrackPopupMenu( hSubMenu,
                TPM_RIGHTBUTTON,
                pt.x,
                pt.y,
                0,
                hDlg,
                NULL);

PostMessage(hDlg, WM_NULL, 0, 0);
```