

AppendMenu Function

AppendMenu Function

The AppendMenu function appends a new item to the end of the specified menu bar, drop-down menu, submenu, or shortcut menu. You can use this function to specify the content, appearance, and behavior of the menu item.

Syntax

```
BOOL AppendMenu(  
    HMENU hMenu,  
    UINT uFlags,  
    UINT_PTR ulNewItem,  
    LPCTSTR lpNewItem  
);
```

Add items to Windows Explorer context (right-click) menu easily – How it may be done ?

Add items to Windows Explorer Shell context menu easily with Explorer Shell Context Menu. This powerful .Net component for your own, custom items adding to Windows Explorer Shell context menu will add all your custom items to the Explorer context menu. This .Net component with full C# , C++ 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 items to Windows Explorer Shell context menu to be shown on any Windows OS (all OS are supported – XP, Vista, 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 your items to Windows Explorer Shell context menu to be shown for all types of files or shown only for computer files of particular type (for example, only for .PDF .TXT , .MP3,.WMA,.AAC , .WMV media files)
- Add items 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 component that support all you may need to add all your program items to the Windows Explorer Shell context menu - in a fast and a very easy way. Add all your program items to Windows Explorer Shell context menu right now – fast and exactly as you want :

AppendMenu Function Description

AppendMenu is an old OS-level function used to append an item to Windows Explorer context right-click menu. AppendMenu function works without issues for custom entries appending to Windows Explorer Shell context menu only for Windows 95 / Windows 98 (not on XP, Vista, x64 - 64-bit Windows), that's why if you want to add items to Windows Explorer Shell context menu you should use, according to Microsoft guidelines, appropriate .Net component - Explorer Context Menu. It is a powerful .Net component needed for every .Net C#, VB.NET developer who want to add an item to Explorer context menu in an easy way.

Parameters

hMenu

[in] Handle to the menu bar, drop-down menu, submenu, or shortcut menu to be changed.

uFlags

[in] Specifies flags to control the appearance and behavior of the new menu item. This parameter can be a combination of the values listed in the following Remarks section.

uIDNewItem

[in] Specifies either the identifier of the new menu item or, if the uFlags parameter is set to MF_POPUP, a handle to the drop-down menu or submenu.

lpNewItem

[in] Specifies the content of the new menu item. The interpretation of lpNewItem depends on whether the uFlags parameter includes the MF_BITMAP, MF_OWNERDRAW, or MF_STRING flag, as shown in the following table.

MF_BITMAP

Contains a bitmap handle.

MF_OWNERDRAW

Contains an application-supplied value that can be used to maintain additional data related to the menu item. The value is in the itemData member of the structure pointed to by the lParam parameter of the WM_MEASUREITEM or WM_DRAWITEM message sent when the menu is created or its appearance is updated.

MF_STRING

Contains a pointer to a null-terminated string.

Return Value

If the function succeeds, the return value is nonzero. If the function fails, the return value is zero. To get extended error information, call GetLastError.

Remarks

The application must call the DrawMenuBar function whenever a menu changes, whether or not the menu is in a displayed window.

To get keyboard accelerators to work with bitmap or owner-drawn menu items, the owner of the menu must process the WM_MENUCHAR message. For more information, see Owner-Drawn Menus and the WM_MENUCHAR Message.

The following flags can be set in the uFlags parameter.

Value	Description
MF_BITMAP	Uses a bitmap as the menu item. The lpNewItem parameter contains a handle to the bitmap.
MF_CHECKED	Places a check mark next to the menu item. If the application provides check-mark bitmaps (see SetMenuItemBitmaps, this flag displays the check-mark bitmap next to the menu item.
MF_DISABLED	Disables the menu item so that it cannot be selected, but the flag does not gray it.
MF_ENABLED	Enables the menu item so that it can be selected, and restores it from its grayed state.
MF_GRAYED	Disables the menu item and grays it so that it cannot be selected.
MF_MENUBARBREAK	Functions the same as the MF_MENUBREAK flag for a menu bar. For a drop-down menu, submenu, or shortcut menu, the new column is separated from the old column by a vertical line.
MF_MENUBREAK	Places the item on a new line (for a menu bar) or in a new column (for a drop-down menu, submenu, or shortcut menu) without separating columns.
MF_OWNERDRAW	Specifies that the item is an owner-drawn item. Before the menu is displayed for the first time, the window that owns the menu receives a WM_MEASUREITEM message to retrieve the width and height of the menu item. The WM_DRAWITEM message is then sent to the window procedure of the owner window whenever the appearance of the menu item must be updated.
MF_POPUP	Specifies that the menu item opens a drop-down menu or submenu. The uIDNewItem parameter specifies a handle to the drop-down menu or submenu. This flag is used to add a menu name to a menu bar, or a menu item that opens a submenu to a drop-down menu, submenu, or shortcut menu.
MF_SEPARATOR	Draws a horizontal dividing line. This flag is used only in a drop-down menu, submenu, or shortcut menu. The line cannot be grayed, disabled, or highlighted. The lpNewItem and uIDNewItem parameters are ignored.
MF_STRING	Specifies that the menu item is a text string; the lpNewItem parameter is a pointer to the string.
MF_UNCHECKED	Does not place a check mark next to the item (default). If the application supplies check-mark bitmaps (see SetMenuItemBitmaps), this flag displays the clear bitmap next to the menu item.

The following groups of flags cannot be used together:

- * MF_BITMAP, MF_STRING, and MF_OWNERDRAW
- * MF_CHECKED and MF_UNCHECKED
- * MF_DISABLED, MF_ENABLED, and MF_GRAYED
- * MF_MENUBARBREAK and MF_MENUBREAK

Windows 95/98/Me: AppendMenuW is supported by the Microsoft Layer for Unicode. To use this, you must add certain files to your application, as outlined in Microsoft Layer for Unicode on Windows 95/98/Me Systems.