The following predefined system classes can be specified in the *lpClassName* parameter. Note the corresponding control styles you can use in the *dwStyle* parameter.

Expand table

|  |  |
| --- | --- |
| **System class** | **Meaning** |
| **BUTTON** | Designates a small rectangular child window that represents a button the user can click to turn it on or off. Button controls can be used alone or in groups, and they can either be labeled or appear without text. Button controls typically change appearance when the user clicks them. For more information, see [Buttons](https://learn.microsoft.com/en-us/windows/desktop/Controls/buttons)  For a table of the button styles you can specify in the *dwStyle* parameter, see [Button Styles](https://learn.microsoft.com/en-us/windows/desktop/Controls/button-styles). |
| **COMBOBOX** | Designates a control consisting of a list box and a selection field similar to an edit control. When using this style, an application should either display the list box at all times or enable a drop-down list box. If the list box is visible, typing characters into the selection field highlights the first list box entry that matches the characters typed. Conversely, selecting an item in the list box displays the selected text in the selection field.  For more information, see [Combo Boxes](https://learn.microsoft.com/en-us/windows/desktop/Controls/combo-boxes). For a table of the combo box styles you can specify in the *dwStyle* parameter, see [Combo Box Styles](https://learn.microsoft.com/en-us/windows/desktop/Controls/combo-box-styles). |
| **EDIT** | Designates a rectangular child window into which the user can type text from the keyboard. The user selects the control and gives it the keyboard focus by clicking it or moving to it by pressing the TAB key. The user can type text when the edit control displays a flashing caret; use the mouse to move the cursor, select characters to be replaced, or position the cursor for inserting characters; or use the BACKSPACE key to delete characters. For more information, see [Edit Controls](https://learn.microsoft.com/en-us/windows/desktop/Controls/edit-controls).  For a table of the edit control styles you can specify in the *dwStyle* parameter, see [Edit Control Styles](https://learn.microsoft.com/en-us/windows/desktop/Controls/edit-control-styles). |
| **LISTBOX** | Designates a list of character strings. Specify this control whenever an application must present a list of names, such as file names, from which the user can choose. The user can select a string by clicking it. A selected string is highlighted, and a notification message is passed to the parent window. For more information, see [List Boxes](https://learn.microsoft.com/en-us/windows/desktop/uxguide/ctrl-list-boxes).  For a table of the list box styles you can specify in the *dwStyle* parameter, see [List Box Styles](https://learn.microsoft.com/en-us/windows/desktop/Controls/list-box-styles). |
| **MDICLIENT** | Designates an MDI client window. This window receives messages that control the MDI application's child windows. The recommended style bits are **WS\_CLIPCHILDREN** and **WS\_CHILD**. Specify the **WS\_HSCROLL** and **WS\_VSCROLL** styles to create an MDI client window that allows the user to scroll MDI child windows into view.  For more information, see [Multiple Document Interface](https://learn.microsoft.com/en-us/windows/desktop/winmsg/multiple-document-interface). |
| **RichEdit** | Designates a Microsoft Rich Edit 1.0 control. This window lets the user view and edit text with character and paragraph formatting, and can include embedded Component Object Model (COM) objects. For more information, see [Rich Edit Controls](https://learn.microsoft.com/en-us/windows/desktop/Controls/rich-edit-controls).  For a table of the rich edit control styles you can specify in the *dwStyle* parameter, see [Rich Edit Control Styles](https://learn.microsoft.com/en-us/windows/desktop/Controls/rich-edit-control-styles). |
| **RICHEDIT\_CLASS** | Designates a Microsoft Rich Edit 2.0 control. This controls let the user view and edit text with character and paragraph formatting, and can include embedded COM objects. For more information, see [Rich Edit Controls](https://learn.microsoft.com/en-us/windows/desktop/Controls/rich-edit-controls).  For a table of the rich edit control styles you can specify in the *dwStyle* parameter, see [Rich Edit Control Styles](https://learn.microsoft.com/en-us/windows/desktop/Controls/rich-edit-control-styles). |
| **SCROLLBAR** | Designates a rectangle that contains a scroll box and has direction arrows at both ends. The scroll bar sends a notification message to its parent window whenever the user clicks the control. The parent window is responsible for updating the position of the scroll box, if necessary. For more information, see [Scroll Bars](https://learn.microsoft.com/en-us/windows/desktop/Controls/scroll-bars).  For a table of the scroll bar control styles you can specify in the *dwStyle* parameter, see [Scroll Bar Control Styles](https://learn.microsoft.com/en-us/windows/desktop/Controls/scroll-bar-control-styles). |
| **STATIC** | Designates a simple text field, box, or rectangle used to label, box, or separate other controls. Static controls take no input and provide no output. For more information, see [Static Controls](https://learn.microsoft.com/en-us/windows/desktop/Controls/static-controls).  For a table of the static control styles you can specify in the *dwStyle* parameter, see [Static Control Styles](https://learn.microsoft.com/en-us/windows/desktop/Controls/static-control-styles). |

**CreateWindow** is implemented as a call to the [CreateWindowEx](https://learn.microsoft.com/en-us/windows/desktop/api/winuser/nf-winuser-createwindowexa) function, as shown below.

syntaxCopy

#define CreateWindowA(lpClassName, lpWindowName, dwStyle, x, y, nWidth, nHeight, hWndParent, hMenu, hInstance, lpParam)\

CreateWindowExA(0L, lpClassName, lpWindowName, dwStyle, x, y, nWidth, nHeight, hWndParent, hMenu, hInstance, lpParam)

#define CreateWindowW(lpClassName, lpWindowName, dwStyle, x, y, nWidth, nHeight, hWndParent, hMenu, hInstance, lpParam)\

CreateWindowExW(0L, lpClassName, lpWindowName, dwStyle, x, y, nWidth, nHeight, hWndParent, hMenu, hInstance, lpParam)

#ifdef UNICODE

#define CreateWindow CreateWindowW

#else

#define CreateWindow CreateWindowA

#endif

As in [CreateWindowW macro (winuser.h) - Win32 apps | Microsoft Learn](https://learn.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-createwindoww?devlangs=cpp&f1url=%3FappId%3DDev17IDEF1%26l%3DEN-US%26k%3Dk(WINUSER%252FCreateWindowW)%3Bk(CreateWindowW)%3Bk(CreateWindow)%3Bk(DevLang-C%252B%252B)%3Bk(TargetOS-Windows)%26rd%3Dtrue)