

Contents

Winuser.h

Overview

ACCEL structure

ACCESSTIMEOUT structure

ActivateKeyboardLayout function

AddClipboardFormatListener function

AdjustWindowRect function

AdjustWindowRectEx function

AdjustWindowRectExForDpi function

AllowSetForegroundWindow function

ALTTABINFO structure

AnimateWindow function

ANIMATIONINFO structure

AnyPopup function

AppendMenuA function

AppendMenuW function

AR_STATE enumeration

AreDpiAwarenessContextsEqual function

ArrangeIconicWindows function

AttachThreadInput function

AUDIODESCRIPTION structure

BeginDeferWindowPos function

BeginPaint function

BlockInput function

BringWindowToTop function

BroadcastSystemMessage function

BroadcastSystemMessageExA function

BroadcastSystemMessageExW function

BroadcastSystemMessageW function

BSMINFO structure
CalculatePopupWindowPosition function
CallMsgFilterA function
CallMsgFilterW function
CallNextHookEx function
CallWindowProcA function
CallWindowProcW function
CascadeWindows function
CBT_CREATEWNTDA structure
CBT_CREATEWNDW structure
CBTACTIVATESTRUCT structure
ChangeClipboardChain function
ChangeDisplaySettingsA function
ChangeDisplaySettingsExA function
ChangeDisplaySettingsExW function
ChangeDisplaySettingsW function
CHANGEFILTERSTRUCT structure
ChangeWindowMessageFilter function
ChangeWindowMessageFilterEx function
CharLowerA function
CharLowerBuffA function
CharLowerBuffW function
CharLowerW function
CharNextA function
CharNextExA function
CharNextW function
CharPrevA function
CharPrevExA function
CharPrevW function
CharToOemA function
CharToOemBuffA function
CharToOemBuffW function

CharToOemW function
CharUpperA function
CharUpperBuffA function
CharUpperBuffW function
CharUpperW function
CheckDlgButton function
CheckMenuItem function
CheckMenuRadioItem function
CheckRadioButton function
ChildWindowFromPoint function
ChildWindowFromPointEx function
CLIENTCREATESTRUCT structure
ClientToScreen function
ClipCursor function
CloseClipboard function
CloseDesktop function
CloseGestureInfoHandle function
CloseTouchInputHandle function
CloseWindow function
CloseWindowStation function
COMBOBOXINFO structure
COMPAREITEMSTRUCT structure
CopyAcceleratorTableA function
CopyAcceleratorTableW function
CopyCursor macro
COPYDATASTRUCT structure
CopyIcon function
CopyImage function
CopyRect function
CountClipboardFormats function
CreateAcceleratorTableA function
CreateAcceleratorTableW function

CreateCaret function
CreateCursor function
CreateDesktopA function
CreateDesktopExA function
CreateDesktopExW function
CreateDesktopW function
CreateDialogA macro
CreateDialogIndirectA macro
CreateDialogIndirectParamA function
CreateDialogIndirectParamW function
CreateDialogIndirectW macro
CreateDialogParamA function
CreateDialogParamW function
CreateDialogW macro
CreateIcon function
CreateIconFromResource function
CreateIconFromResourceEx function
CreateIconIndirect function
CreateMDIWindowA function
CreateMDIWindowW function
CreateMenu function
CreatePopupMenu function
CREATESTRUCTA structure
CREATESTRUCTW structure
CreateSyntheticPointerDevice function
CreateWindowA macro
CreateWindowExA function
CreateWindowExW function
CreateWindowStationA function
CreateWindowStationW function
CreateWindowW macro
CURSORINFO structure

CURSORSHAPE structure
CWPRETSTRUCT structure
CWPSTRUCT structure
DEBUGHOOKINFO structure
DefDlgProcW function
DeferWindowPos function
DefFrameProcA function
DefFrameProcW function
DefMDIChildProcA function
DefMDIChildProcW function
DefRawInputProc function
DefWindowProcA function
DefWindowProcW function
DELETEITEMSTRUCT structure
DeleteMenu function
DeregisterShellHookWindow function
DestroyAcceleratorTable function
DestroyCaret function
DestroyCursor function
DestroyIcon function
DestroyMenu function
DestroySyntheticPointerDevice function
DestroyWindow function
DIALOG_CONTROL_DPI_CHANGE_BEHAVIORS enumeration
DIALOG_DPI_CHANGE_BEHAVIORS enumeration
DialogBoxA macro
DialogBoxIndirectA macro
DialogBoxIndirectParamA function
DialogBoxIndirectParamW function
DialogBoxIndirectW macro
DialogBoxParamA function
DialogBoxParamW function

DialogBoxW macro
DisableProcessWindowsGhosting function
DispatchMessage function
DispatchMessageA function
DispatchMessageW function
DisplayConfigGetDeviceInfo function
DisplayConfigSetDeviceInfo function
DlgDirListA function
DlgDirListComboBoxA function
DlgDirListComboBoxW function
DlgDirListW function
DlgDirSelectComboBoxExA function
DlgDirSelectComboBoxExW function
DlgDirSelectExA function
DlgDirSelectExW function
DLGITEMTEMPLATE structure
DLGPROC callback function
DLGTEMPLATE structure
DragDetect function
DrawAnimatedRects function
DrawCaption function
DrawEdge function
DrawFocusRect function
DrawFrameControl function
DrawIcon function
DrawIconEx function
DRAWITEMSTRUCT structure
DrawMenuBar function
DrawStateA function
DRAWSTATEPROC callback function
DrawStateW function
DrawText function

DrawTextA function
DrawTextExA function
DrawTextExW function
DRAWTEXTPARAMS structure
DrawTextW function
EDITWORDBREAKPROCA callback function
EDITWORDBREAKPROCW callback function
EmptyClipboard function
EnableMenuItem function
EnableMouseInPointer function
EnableNonClientDpiScaling function
EnableScrollBar function
EnableWindow function
EndDeferWindowPos function
EndDialog function
EndMenu function
EndPaint function
EndTask function
EnumChildWindows function
EnumClipboardFormats function
EnumDesktopsA function
EnumDesktopsW function
EnumDesktopWindows function
EnumDisplayDevicesA function
EnumDisplayDevicesW function
EnumDisplayMonitors function
EnumDisplaySettingsA function
EnumDisplaySettingsExA function
EnumDisplaySettingsExW function
EnumDisplaySettingsW function
EnumPropsA function
EnumPropsExA function

EnumPropsExW function
EnumPropsW function
EnumThreadWindows function
EnumWindows function
EnumWindowStationsA function
EnumWindowStationsW function
EqualRect function
EvaluateProximityToPolygon function
EvaluateProximityToRect function
EVENTMSG structure
ExcludeUpdateRgn function
ExitWindows macro
ExitWindowsEx function
FEEDBACK_TYPE enumeration
FillRect function
FILTERKEYS structure
FindWindowA function
FindWindowExA function
FindWindowExW function
FindWindowW function
FlashWindow function
FlashWindowEx function
FLASHWINFO structure
FrameRect function
GESTURECONFIG structure
GESTUREINFO structure
GESTURENOTIFYSTRUCT structure
GET_APPCOMMAND_LPAMM macro
GET_DEVICE_LPAMM macro
GET_FLAGS_LPAMM macro
GET_KEYSTATE_LPAMM macro
GET_KEYSTATE_WPARAM macro

GET_NCHITTEST_WPARAM macro
GET_POINTERID_WPARAM macro
GET_RAWINPUT_CODE_WPARAM macro
GET_WHEEL_DELTA_WPARAM macro
GET_XBUTTON_WPARAM macro
GetActiveWindow function
GetAltTabInfoA function
GetAltTabInfoW function
GetAncestor function
GetAsyncKeyState function
GetAutoRotationState function
GetAwarenessFromDpiAwarenessContext function
GetCapture function
GetCaretBlinkTime function
GetCaretPos function
GetCIMSSM function
GetClassInfoA function
GetClassInfoExA function
GetClassInfoExW function
GetClassInfoW function
GetClassLongA function
GetClassLongPtrA function
GetClassLongPtrW function
GetClassLongW function
GetClassName function
GetClassNameA function
GetClassNameW function
GetClassWord function
GetClientRect function
GetClipboardData function
GetClipboardFormatNameA function
GetClipboardFormatNameW function

GetClipboardOwner function
GetClipboardSequenceNumber function
GetClipboardViewer function
GetClipCursor function
GetComboBoxInfo function
GetCurrentInputMessageSource function
GetCursor function
GetCursorInfo function
GetCursorPos function
GetDC function
GetDCEX function
GetDesktopWindow function
GetDialogBaseUnits function
GetDialogControlDpiChangeBehavior function
GetDialogDpiChangeBehavior function
GetDisplayAutoRotationPreferences function
GetDisplayAutoRotationPreferencesByProcessId function
GetDisplayConfigBufferSizes function
GetDlgCtrlID function
GetDlgItem function
GetDlgItemInt function
GetDlgItemTextA function
GetDlgItemTextW function
GetDoubleClickTime function
GetDpiForSystem function
GetDpiForWindow function
GetDpiFromDpiAwarenessContext function
GetFocus function
GetForegroundWindow function
GetGestureConfig function
GetGestureExtraArgs function
GetGestureInfo function

GetGuiResources function
GetGUIThreadInfo function
GetIconInfo function
GetIconInfoExA function
GetIconInfoExW function
GetInputState function
GetKBCodePage function
GetKeyboardLayout function
GetKeyboardLayoutList function
GetKeyboardLayoutNameA function
GetKeyboardLayoutNameW function
GetKeyboardState function
GetKeyboardType function
GetKeyNameTextA function
GetKeyNameTextW function
GetKeyState function
GetLastActivePopup function
GetLastInputInfo function
GetLayeredWindowAttributes function
GetListBoxInfo function
GetMenu function
GetMenuBarInfo function
GetMenuCheckMarkDimensions function
GetMenuContextHelpId function
GetMenuDefaultItem function
GetMenuInfo function
GetMenuItemCount function
GetMenuItemID function
GetMenuItemInfoA function
GetMenuItemInfoW function
GetMenuItemRect function
GetMenuState function

GetMenuStringA function
GetMenuStringW function
GetMessage function
GetMessageA function
GetMessageExtraInfo function
GetMessagePos function
GetMessageTime function
GetMessageW function
GetMonitorInfoA function
GetMonitorInfoW function
GetMouseMovePointsEx function
GetNextDlgGroupItem function
GetNextDlgTabItem function
GetNextWindow macro
GetOpenClipboardWindow function
GetParent function
GetPhysicalCursorPos function
GetPointerCursorId function
GetPointerDevice function
GetPointerDeviceCursors function
GetPointerDeviceProperties function
GetPointerDeviceRects function
GetPointerDevices function
GetPointerFrameInfo function
GetPointerFrameInfoHistory function
GetPointerFramePenInfo function
GetPointerFramePenInfoHistory function
GetPointerFrameTouchInfo function
GetPointerFrameTouchInfoHistory function
GetPointerInfo function
GetPointerInfoHistory function
GetPointerInputTransform function

GetPointerPenInfo function
GetPointerPenInfoHistory function
GetPointerTouchInfo function
GetPointerTouchInfoHistory function
GetPointerType function
GetPriorityClipboardFormat function
GetProcessDefaultLayout function
GetProcessWindowStation function
GetPropA function
GetPropW function
GetQueueStatus function
GetRawInputBuffer function
GetRawInputData function
GetRawInputDeviceInfoA function
GetRawInputDeviceInfoW function
GetRawInputDeviceList function
GetRawPointerDeviceData function
GetRegisteredRawInputDevices function
GetScrollBarInfo function
GetScrollInfo function
GetScrollPos function
GetScrollRange function
GetShellWindow function
GetSubMenu function
GetSysColor function
GetSysColorBrush function
GetSystemDpiForProcess function
GetSystemMenu function
GetSystemMetrics function
GetSystemMetricsForDpi function
GetTabbedTextExtentA function
GetTabbedTextExtentW function

GetThreadDesktop function
GetThreadDpiAwarenessContext function
GetThreadDpiHostingBehavior function
GetTitleBarInfo function
GetTopWindow function
GetTouchInputInfo function
GetUnpredictedMessagePos function
GetUpdatedClipboardFormats function
GetUpdateRect function
GetUpdateRgn function
GetUserObjectInformationA function
GetUserObjectInformationW function
GetUserObjectSecurity function
GetWindow function
GetWindowContextHelpId function
GetWindowDC function
GetWindowDisplayAffinity function
GetWindowDpiAwarenessContext function
GetWindowDpiHostingBehavior function
GetWindowFeedbackSetting function
GetWindowInfo function
GetWindowLongA function
GetWindowLongPtrA function
GetWindowLongPtrW function
GetWindowLongW function
GetWindowModuleFileNameA function
GetWindowModuleFileNameW function
GetWindowPlacement function
GetWindowRect function
GetWindowRgn function
GetWindowRgnBox function
GetWindowTextA function

GetWindowTextLengthA function
GetWindowTextLengthW function
GetWindowTextW function
GetWindowThreadProcessId function
GID_ROTATE_ANGLE_FROM_ARGUMENT macro
GID_ROTATE_ANGLE_TO_ARGUMENT macro
GrayStringA function
GRAYSTRINGPROC callback function
GrayStringW function
GUI_THREAD_INFO structure
HARDWAREINPUT structure
HAS_POINTER_CONFIDENCE_WPARAM macro
HELPINFO structure
HELPWININFOA structure
HELPWININFOW structure
HideCaret function
HIGHCONTRASTA structure
HIGHCONTRASTW structure
HiliteMenuItem function
HOOKPROC callback function
ICONINFO structure
ICONINFOEXA structure
ICONINFOEXW structure
ICONMETRICS_A structure
ICONMETRICS_W structure
InflateRect function
InitializeTouchInjection function
InjectSyntheticPointerInput function
InjectTouchInput function
INPUT structure
INPUT_INJECTION_VALUE structure
INPUT_MESSAGE_DEVICE_TYPE enumeration

INPUT_MESSAGE_ORIGIN_ID enumeration
INPUT_MESSAGE_SOURCE structure
INPUT_TRANSFORM structure
InSendMessage function
InSendMessageEx function
InsertMenuA function
InsertMenuItemA function
InsertMenuItemW function
InsertMenuW function
InternalGetWindowText function
IntersectRect function
InvalidateRect function
InvalidateRgn function
InvertRect function
IS_INTRESOURCE macro
IS_POINTER_CANCELED_WPARAM macro
IS_POINTER_FIFTHBUTTON_WPARAM macro
IS_POINTER_FIRSTBUTTON_WPARAM macro
IS_POINTER_FLAG_SET_WPARAM macro
IS_POINTER_FOURTHBUTTON_WPARAM macro
IS_POINTER_INCONTACT_WPARAM macro
IS_POINTER_INRANGE_WPARAM macro
IS_POINTER_NEW_WPARAM macro
IS_POINTER_SECONDBUTTON_WPARAM macro
IS_POINTER_THIRDBUTTON_WPARAM macro
IsCharAlphaA function
IsCharAlphaNumericA function
IsCharAlphaNumericW function
IsCharAlphaW function
IsCharLowerA function
IsCharLowerW function
IsCharUpperA function

IsCharUpperW function
IsChild function
IsClipboardFormatAvailable function
IsDialogMessageA function
IsDialogMessageW function
IsDlgButtonChecked function
IsGUIThread function
IsHungAppWindow function
IsIconic function
IsImmersiveProcess function
IsMenu function
IsMouseInPointerEnabled function
IsProcessDPIAware function
IsRectEmpty function
IsTouchWindow function
IsValidDpiAwarenessContext function
IsWindow function
IsWindowEnabled function
IsWindowUnicode function
IsWindowVisible function
IsWinEventHookInstalled function
IsWow64Message function
IsZoomed function
KBDLLHOOKSTRUCT structure
keybd_event function
KEYBDINPUT structure
KillTimer function
LASTINPUTINFO structure
LoadAcceleratorsA function
LoadAcceleratorsW function
LoadBitmapA function
LoadBitmapW function

LoadCursorA function
LoadCursorFromFileA function
LoadCursorFromFileW function
LoadCursorW function
LoadIconA function
LoadIconW function
LoadImageA function
LoadImageW function
LoadKeyboardLayoutA function
LoadKeyboardLayoutW function
LoadMenuA function
LoadMenuIndirectA function
LoadMenuIndirectW function
LoadMenuW function
LoadStringA function
LoadStringW function
LockSetForegroundWindow function
LockWindowUpdate function
LockWorkStation function
LogicalToPhysicalPoint function
LogicalToPhysicalPointForPerMonitorDPI function
LookupIconIdFromDirectory function
LookupIconIdFromDirectoryEx function
MAKEINTRESOURCEA macro
MAKEINTRESOURCEW macro
MAKELPARAM macro
MAKELRESULT macro
MAKEWPARAM macro
MapDialogRect function
MapVirtualKeyA function
MapVirtualKeyExA function
MapVirtualKeyExW function

MapVirtualKeyW function
MapWindowPoints function
MDICREATESTRUCTA structure
MDICREATESTRUCTW structure
MDINEXTMENU structure
MEASUREITEMSTRUCT structure
MENUBARINFO structure
MENUGETOBJECTINFO structure
MENUINFO structure
MenuItemFromPoint function
MENUITEMINFOA structure
MENUITEMINFOW structure
MENUITEMTEMPLATE structure
MENUITEMTEMPLATEHEADER structure
MessageBeep function
MessageBox function
MessageBoxA function
MessageBoxExA function
MessageBoxExW function
MessageBoxIndirectA function
MessageBoxIndirectW function
MessageBoxW function
MINIMIZEDMETRICS structure
MINMAXINFO structure
ModifyMenuA function
ModifyMenuW function
MONITORENUMPROC callback function
MonitorFromPoint function
MonitorFromRect function
MonitorFromWindow function
MONITORINFO structure
MONITORINFOEXA structure

MONITORINFOEXW structure
mouse_event function
MOUSEHOOKSTRUCT structure
MOUSEHOOKSTRUCTEX structure
MOUSEINPUT structure
MOUSEKEYS structure
MOUSEMOVEPOINT structure
MoveWindow function
MSG structure
MSGBOXPARAMSA structure
MSGBOXPARAMSW structure
MsgWaitForMultipleObjects function
MsgWaitForMultipleObjectsEx function
MSLLHOOKSTRUCT structure
MULTIKEYHELPA structure
MULTIKEYHELPW structure
NCCALCSIZE_PARAMS structure
NEXTRAWINPUTBLOCK macro
NMHDR structure
NONCLIENTMETRICS_SA structure
NONCLIENTMETRICS_SW structure
NotifyWinEvent function
OemKeyScan function
OemToCharA function
OemToCharBuffA function
OemToCharBuffW function
OemToCharW function
OffsetRect function
OpenClipboard function
OpenDesktopA function
OpenDesktopW function
OpenIcon function

OpenInputDesktop function
OpenWindowStationA function
OpenWindowStationW function
ORIENTATION_PREFERENCE enumeration
PackTouchHitTestingProximityEvaluation function
PaintDesktop function
PAINTSTRUCT structure
PeekMessageA function
PeekMessageW function
PhysicalToLogicalPoint function
PhysicalToLogicalPointForPerMonitorDPI function
POINTER_BUTTON_CHANGE_TYPE enumeration
POINTER_DEVICE_CURSOR_INFO structure
POINTER_DEVICE_CURSOR_TYPE enumeration
POINTER_DEVICE_INFO structure
POINTER_DEVICE_PROPERTY structure
POINTER_DEVICE_TYPE enumeration
POINTER_FEEDBACK_MODE enumeration
POINTER_INFO structure
POINTER_PEN_INFO structure
POINTER_TOUCH_INFO structure
POINTER_TYPE_INFO structure
POINTSTOPOINT macro
POINTTOPOINTS macro
PostMessageA function
PostMessageW function
PostQuitMessage function
PostThreadMessageA function
PostThreadMessageW function
POWERBROADCAST_SETTING structure
PrintWindow function
PrivateExtractIconsA function

PrivateExtractIconsW function
PROPENUMPROCA callback function
PROPENUMPROCEXA callback function
PROPENUMPROCEXW callback function
PROPENUMPROCW callback function
PtInRect function
QueryDisplayConfig function
RAWHID structure
RAWINPUT structure
RAWINPUTDEVICE structure
RAWINPUTDEVICELIST structure
RAWINPUTHEADER structure
RAWKEYBOARD structure
RAWMOUSE structure
RealChildWindowFromPoint function
RealGetWindowClassW function
RedrawWindow function
RegisterClassA function
RegisterClassExA function
RegisterClassExW function
RegisterClassW function
RegisterClipboardFormatA function
RegisterClipboardFormatW function
RegisterDeviceNotificationA function
RegisterDeviceNotificationW function
RegisterHotKey function
RegisterPointerDeviceNotifications function
RegisterPointerInputTarget function
RegisterPointerInputTargetEx function
RegisterPowerSettingNotification function
RegisterRawInputDevices function
RegisterShellHookWindow function

RegisterSuspendResumeNotification function

RegisterTouchHitTestingWindow function

RegisterTouchWindow function

RegisterWindowMessageA function

RegisterWindowMessageW function

ReleaseCapture function

ReleaseDC function

RemoveClipboardFormatListener function

RemoveMenu function

RemovePropA function

RemovePropW function

ReplyMessage function

RID_DEVICE_INFO structure

RID_DEVICE_INFO_HID structure

RID_DEVICE_INFO_KEYBOARD structure

RID_DEVICE_INFO_MOUSE structure

ScreenToClient function

SCROLLBARINFO structure

ScrollDC function

SCROLLINFO structure

ScrollWindow function

ScrollWindowEx function

SENDASYNCPROC callback function

SendDlgItemMessageA function

SendDlgItemMessageW function

SendInput function

SendMessage function

SendMessageA function

SendMessageCallbackA function

SendMessageCallbackW function

SendMessageTimeoutA function

SendMessageTimeoutW function

SendMessageW function
SendNotifyMessageA function
SendNotifyMessageW function
SERIALKEYSA structure
SERIALKEYSW structure
SetActiveWindow function
SetCapture function
SetCaretBlinkTime function
SetCaretPos function
SetClassLongA function
SetClassLongPtrA function
SetClassLongPtrW function
SetClassLongW function
SetClassWord function
SetClipboardData function
SetClipboardViewer function
SetCoalescableTimer function
SetCursor function
SetCursorPos function
SetDialogControlDpiChangeBehavior function
SetDialogDpiChangeBehavior function
SetDisplayAutoRotationPreferences function
SetDisplayConfig function
SetDlgItemInt function
SetDlgItemTextA function
SetDlgItemTextW function
SetDoubleClickTime function
SetFocus function
SetForegroundWindow function
SetGestureConfig function
SetKeyboardState function
SetLastErrorEx function

SetLayeredWindowAttributes function
SetMenu function
SetMenuContextHelpId function
SetMenuDefaultItem function
SetMenuInfo function
SetMenuItemBitmaps function
SetMenuItemInfoA function
SetMenuItemInfoW function
SetMessageExtraInfo function
SetParent function
SetPhysicalCursorPos function
SetProcessDefaultLayout function
SetProcessDPIAware function
SetProcessDpiAwarenessContext function
SetProcessRestrictionExemption function
SetProcessWindowStation function
SetPropA function
SetPropW function
SetRect function
SetRectEmpty function
SetScrollInfo function
SetScrollPos function
SetScrollRange function
SetSysColors function
SetSystemCursor function
SetThreadDesktop function
SetThreadDpiAwarenessContext function
SetThreadDpiHostingBehavior function
SetTimer function
SetUserObjectInformationA function
SetUserObjectInformationW function
SetUserObjectSecurity function

SetWindowContextHelpId function
SetWindowDisplayAffinity function
SetWindowFeedbackSetting function
SetWindowLongA function
SetWindowLongPtrA function
SetWindowLongPtrW function
SetWindowLongW function
SetWindowPlacement function
SetWindowPos function
SetWindowRgn function
SetWindowsHookExA function
SetWindowsHookExW function
SetWindowTextA function
SetWindowTextW function
SetWinEventHook function
ShowCaret function
ShowCursor function
ShowOwnedPopups function
ShowScrollBar function
ShowWindow function
ShowWindowAsync function
ShutdownBlockReasonCreate function
ShutdownBlockReasonDestroy function
ShutdownBlockReasonQuery function
SkipPointerFrameMessages function
SoundSentry function
SOUNDSENTRYA structure
SOUNDSENTRYW structure
STICKYKEYS structure
STYLESTRUCT structure
SubtractRect function
SwapMouseButton function

SwitchDesktop function
SwitchToThisWindow function
SystemParametersInfoA function
SystemParametersInfoForDpi function
SystemParametersInfoW function
TabbedTextOutA function
TabbedTextOutW function
tagPOINTER_INPUT_TYPE enumeration
TileWindows function
TIMERPROC callback function
TITLEBARINFO structure
TITLEBARINFOEX structure
ToAscii function
ToAsciiEx function
TOGGLEKEYS structure
TOUCH_COORD_TO_PIXEL macro
TOUCH_HIT_TESTING_INPUT structure
TOUCH_HIT_TESTING_PROXIMITY_EVALUATION structure
TOUCHINPUT structure
TOUCHPREDICTIONPARAMETERS structure
ToUnicode function
ToUnicodeEx function
TPMPARAMS structure
TrackMouseEvent function
TRACKMOUSEEVENT structure
TrackPopupMenu function
TrackPopupMenuEx function
TranslateAcceleratorA function
TranslateAcceleratorW function
TranslateMDISysAccel function
TranslateMessage function
UnhookWindowsHookEx function

UnhookWinEvent function
UnionRect function
UnloadKeyboardLayout function
UnregisterClassA function
UnregisterClassW function
UnregisterDeviceNotification function
UnregisterHotKey function
UnregisterPointerInputTarget function
UnregisterPointerInputTargetEx function
UnregisterPowerSettingNotification function
UnregisterSuspendResumeNotification function
UnregisterTouchWindow function
UpdateLayeredWindow function
UPDATELAYEREDWINDOWINFO structure
UpdateWindow function
USAGE_PROPERTIES structure
UserHandleGrantAccess function
USEROBJECTFLAGS structure
ValidateRect function
ValidateRgn function
VkKeyScanA function
VkKeyScanExA function
VkKeyScanExW function
VkKeyScanW function
WaitForInputIdle function
WaitMessage function
WindowFromDC function
WindowFromPhysicalPoint function
WindowFromPoint function
WINDOWINFO structure
WINDOWPLACEMENT structure
WINDOWPOS structure

WINEVENTPROC callback function

WinHelpA function

WinHelpW function

WNDCLASSA structure

WNDCLASSEXA structure

WNDCLASSEXW structure

WNDCLASSW structure

wsprintfA function

wsprintfW function

WTSSSESSION_NOTIFICATION structure

wvsprintfA function

wvsprintfW function

winuser.h header

2/1/2021 • 78 minutes to read • [Edit Online](#)

This header is used by Windows Controls. For more information, see:

- [Windows Controls](#) winuser.h contains the following programming interfaces:

Functions

TITLE	DESCRIPTION
ActivateKeyboardLayout	Sets the input locale identifier (formerly called the keyboard layout handle) for the calling thread or the current process. The input locale identifier specifies a locale as well as the physical layout of the keyboard.
AddClipboardFormatListener	Places the given window in the system-maintained clipboard format listener list.
AdjustWindowRect	Calculates the required size of the window rectangle, based on the desired client-rectangle size. The window rectangle can then be passed to the <code>CreateWindow</code> function to create a window whose client area is the desired size.
AdjustWindowRectEx	Calculates the required size of the window rectangle, based on the desired size of the client rectangle. The window rectangle can then be passed to the <code>CreateWindowEx</code> function to create a window whose client area is the desired size.
AdjustWindowRectExForDpi	Calculates the required size of the window rectangle, based on the desired size of the client rectangle and the provided DPI.
AllowSetForegroundWindow	Enables the specified process to set the foreground window using the <code>SetForegroundWindow</code> function. The calling process must already be able to set the foreground window. For more information, see <code>Remarks</code> later in this topic.
AnimateWindow	Enables you to produce special effects when showing or hiding windows. There are four types of animation: <code>_roll</code> , <code>slide</code> , <code>collapse</code> or <code>expand</code> , and <code>alpha-blended fade</code> .
AnyPopup	Indicates whether an owned, visible, top-level pop-up, or overlapped window exists on the screen. The function searches the entire screen, not just the calling application's client area.
AppendMenuA	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.


TITLE	DESCRIPTION
AppendMenuW	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.
AreDpiAwarenessContextsEqual	Determines whether two DPI_AWARENESS_CONTEXT values are identical.
ArrangeIconicWindows	Arranges all the minimized (iconic) child windows of the specified parent window.
AttachThreadInput	Attaches or detaches the input processing mechanism of one thread to that of another thread.
BeginDeferWindowPos	Allocates memory for a multiple-window- position structure and returns the handle to the structure.
BeginPaint	The BeginPaint function prepares the specified window for painting and fills a PAINTSTRUCT structure with information about the painting.
BlockInput	Blocks keyboard and mouse input events from reaching applications.
BringWindowToTop	Brings the specified window to the top of the Z order. If the window is a top-level window, it is activated. If the window is a child window, the top-level parent window associated with the child window is activated.
BroadcastSystemMessage	Sends a message to the specified recipients.
BroadcastSystemMessageExA	Sends a message to the specified recipients.
BroadcastSystemMessageExW	Sends a message to the specified recipients.
BroadcastSystemMessageW	Sends a message to the specified recipients.
CalculatePopupWindowPosition	Calculates an appropriate pop-up window position using the specified anchor point, pop-up window size, flags, and the optional exclude rectangle.
CallMsgFilterA	Passes the specified message and hook code to the hook procedures associated with the WH_SYSMSGFILTER and WH_MSGFILTER hooks.
CallMsgFilterW	Passes the specified message and hook code to the hook procedures associated with the WH_SYSMSGFILTER and WH_MSGFILTER hooks.
CallNextHookEx	Passes the hook information to the next hook procedure in the current hook chain. A hook procedure can call this function either before or after processing the hook information.

TITLE	DESCRIPTION
CallWindowProcA	Passes message information to the specified window procedure.
CallWindowProcW	Passes message information to the specified window procedure.
CascadeWindows	Cascades the specified child windows of the specified parent window.
ChangeClipboardChain	Removes a specified window from the chain of clipboard viewers.
ChangeDisplaySettingsA	The ChangeDisplaySettings function changes the settings of the default display device to the specified graphics mode.
ChangeDisplaySettingsExA	The ChangeDisplaySettingsEx function changes the settings of the specified display device to the specified graphics mode.
ChangeDisplaySettingsExW	The ChangeDisplaySettingsEx function changes the settings of the specified display device to the specified graphics mode.
ChangeDisplaySettingsW	The ChangeDisplaySettings function changes the settings of the default display device to the specified graphics mode.
ChangeWindowMessageFilter	Adds or removes a message from the User Interface Privilege Isolation (UIPI) message filter.
ChangeWindowMessageFilterEx	Modifies the User Interface Privilege Isolation (UIPI) message filter for a specified window.
CharLowerA	Converts a character string or a single character to lowercase. If the operand is a character string, the function converts the characters in place.
CharLowerBuffA	Converts uppercase characters in a buffer to lowercase characters. The function converts the characters in place.
CharLowerBuffW	Converts uppercase characters in a buffer to lowercase characters. The function converts the characters in place.
CharLowerW	Converts a character string or a single character to lowercase. If the operand is a character string, the function converts the characters in place.
CharNextA	Retrieves a pointer to the next character in a string. This function can handle strings consisting of either single- or multi-byte characters.
CharNextExA	Retrieves the pointer to the next character in a string. This function can handle strings consisting of either single- or multi-byte characters.

TITLE	DESCRIPTION
CharNextW	Retrieves a pointer to the next character in a string. This function can handle strings consisting of either single- or multi-byte characters.
CharPrevA	Retrieves a pointer to the preceding character in a string. This function can handle strings consisting of either single- or multi-byte characters.
CharPrevExA	Retrieves the pointer to the preceding character in a string. This function can handle strings consisting of either single- or multi-byte characters.
CharPrevW	Retrieves a pointer to the preceding character in a string. This function can handle strings consisting of either single- or multi-byte characters.
CharToOemA	Translates a string into the OEM-defined character set. Warning Do not use.
CharToOemBuffA	Translates a specified number of characters in a string into the OEM-defined character set.
CharToOemBuffW	Translates a specified number of characters in a string into the OEM-defined character set.
CharToOemW	Translates a string into the OEM-defined character set. Warning Do not use.
CharUpperA	Converts a character string or a single character to uppercase. If the operand is a character string, the function converts the characters in place.
CharUpperBuffA	Converts lowercase characters in a buffer to uppercase characters. The function converts the characters in place.
CharUpperBuffW	Converts lowercase characters in a buffer to uppercase characters. The function converts the characters in place.
CharUpperW	Converts a character string or a single character to uppercase. If the operand is a character string, the function converts the characters in place.
CheckDlgButton	Changes the check state of a button control.
CheckMenuItem	Sets the state of the specified menu item's check-mark attribute to either selected or clear.
CheckMenuRadioItem	Checks a specified menu item and makes it a radio item. At the same time, the function clears all other menu items in the associated group and clears the radio-item type flag for those items.
CheckRadioButton	Adds a check mark to (checks) a specified radio button in a group and removes a check mark from (clears) all other radio buttons in the group.

TITLE	DESCRIPTION
ChildWindowFromPoint	Determines which, if any, of the child windows belonging to a parent window contains the specified point. The search is restricted to immediate child windows. Grandchildren, and deeper descendant windows are not searched.
ChildWindowFromPointEx	Determines which, if any, of the child windows belonging to the specified parent window contains the specified point.
ClientToScreen	The ClientToScreen function converts the client-area coordinates of a specified point to screen coordinates.
ClipCursor	Confines the cursor to a rectangular area on the screen.
CloseClipboard	Closes the clipboard.
CloseDesktop	Closes an open handle to a desktop object.
CloseGestureInfoHandle	Closes resources associated with a gesture information handle.
CloseTouchInputHandle	Closes a touch input handle, frees process memory associated with it, and invalidates the handle.
CloseWindow	Minimizes (but does not destroy) the specified window.
CloseWindowStation	Closes an open window station handle.
CopyAcceleratorTableA	Copies the specified accelerator table. This function is used to obtain the accelerator-table data that corresponds to an accelerator-table handle, or to determine the size of the accelerator-table data.
CopyAcceleratorTableW	Copies the specified accelerator table. This function is used to obtain the accelerator-table data that corresponds to an accelerator-table handle, or to determine the size of the accelerator-table data.
CopyCursor	Copies the specified cursor.
CopyIcon	Copies the specified icon from another module to the current module.
CopyImage	Creates a new image (icon, cursor, or bitmap) and copies the attributes of the specified image to the new one. If necessary, the function stretches the bits to fit the desired size of the new image.
CopyRect	The CopyRect function copies the coordinates of one rectangle to another.

TITLE	DESCRIPTION
CountClipboardFormats	Retrieves the number of different data formats currently on the clipboard.
CreateAcceleratorTableA	Creates an accelerator table.
CreateAcceleratorTableW	Creates an accelerator table.
CreateCaret	Creates a new shape for the system caret and assigns ownership of the caret to the specified window. The caret shape can be a line, a block, or a bitmap.
CreateCursor	Creates a cursor having the specified size, bit patterns, and hot spot.
CreateDesktopA	Creates a new desktop, associates it with the current window station of the calling process, and assigns it to the calling thread.
CreateDesktopExA	Creates a new desktop with the specified heap, associates it with the current window station of the calling process, and assigns it to the calling thread.
CreateDesktopExW	Creates a new desktop with the specified heap, associates it with the current window station of the calling process, and assigns it to the calling thread.
CreateDesktopW	Creates a new desktop, associates it with the current window station of the calling process, and assigns it to the calling thread.
CreateDialogA	Creates a modeless dialog box from a dialog box template resource. The CreateDialog macro uses the CreateDialogParam function.
CreateDialogIndirectA	Creates a modeless dialog box from a dialog box template in memory. The CreateDialogIndirect macro uses the CreateDialogIndirectParam function.
CreateDialogIndirectParamA	Creates a modeless dialog box from a dialog box template in memory.
CreateDialogIndirectParamW	Creates a modeless dialog box from a dialog box template in memory.
CreateDialogIndirectW	Creates a modeless dialog box from a dialog box template in memory. The CreateDialogIndirect macro uses the CreateDialogIndirectParam function.
CreateDialogParamA	Creates a modeless dialog box from a dialog box template resource.
CreateDialogParamW	Creates a modeless dialog box from a dialog box template resource.

TITLE	DESCRIPTION
CreateDialogW	Creates a modeless dialog box from a dialog box template resource. The CreateDialog macro uses the CreateDialogParam function.
CreteIcoN	Creates an icon that has the specified size, colors, and bit patterns.
CreteIcoNFromResource	Creates an icon or cursor from resource bits describing the icon.
CreteIcoNFromResourceEx	Creates an icon or cursor from resource bits describing the icon.
CreteIcoNIndirect	Creates an icon or cursor from an ICONINFO structure.
CreateMDIWindowA	Creates a multiple-document interface (MDI) child window.
CreateMDIWindowW	Creates a multiple-document interface (MDI) child window.
CreateMenu	Creates a menu. The menu is initially empty, but it can be filled with menu items by using the InsertMenuItem, AppendMenu, and InsertMenu functions.
CreatePopupMenu	Creates a drop-down menu, submenu, or shortcut menu.
CreateSyntheticPointerDevice	Configures the pointer injection device for the calling application, and initializes the maximum number of simultaneous pointers that the app can inject.
CreateWindowA	Creates an overlapped, pop-up, or child window.
CreateWindowExA	Creates an overlapped, pop-up, or child window with an extended window style; otherwise, this function is identical to the CreateWindow function.
CreateWindowExW	Creates an overlapped, pop-up, or child window with an extended window style; otherwise, this function is identical to the CreateWindow function.
CreateWindowStationA	Creates a window station object, associates it with the calling process, and assigns it to the current session.
CreateWindowStationW	Creates a window station object, associates it with the calling process, and assigns it to the current session.
CreateWindowW	Creates an overlapped, pop-up, or child window.
DefDlgProcW	Calls the default dialog box window procedure to provide default processing for any window messages that a dialog box with a private window class does not process.
DeferWindowPos	Updates the specified multiple-window  position structure for the specified window.

TITLE	DESCRIPTION
DefFrameProcA	Provides default processing for any window messages that the window procedure of a multiple-document interface (MDI) frame window does not process.
DefFrameProcW	Provides default processing for any window messages that the window procedure of a multiple-document interface (MDI) frame window does not process.
DefMDIChildProcA	Provides default processing for any window message that the window procedure of a multiple-document interface (MDI) child window does not process.
DefMDIChildProcW	Provides default processing for any window message that the window procedure of a multiple-document interface (MDI) child window does not process.
DefRawInputProc	Verifies that the size of the RAWINPUTHEADER structure is correct.
DefWindowProcA	Calls the default window procedure to provide default processing for any window messages that an application does not process.
DefWindowProcW	Calls the default window procedure to provide default processing for any window messages that an application does not process.
DeleteMenu	Deletes an item from the specified menu. If the menu item opens a menu or submenu, this function destroys the handle to the menu or submenu and frees the memory used by the menu or submenu.
DeregisterShellHookWindow	Unregisters a specified Shell window that is registered to receive Shell hook messages.
DestroyAcceleratorTable	Destroys an accelerator table.
DestroyCaret	Destroys the caret's current shape, frees the caret from the window, and removes the caret from the screen.
DestroyCursor	Destroys a cursor and frees any memory the cursor occupied. Do not use this function to destroy a shared cursor.
DestroyIcon	Destroys an icon and frees any memory the icon occupied.
DestroyMenu	Destroys the specified menu and frees any memory that the menu occupies.
DestroySyntheticPointerDevice	Destroys the specified pointer injection device.
DestroyWindow	Destroys the specified window.

TITLE	DESCRIPTION
DialogBoxA	Creates a modal dialog box from a dialog box template resource. DialogBox does not return control until the specified callback function terminates the modal dialog box by calling the EndDialog function.
DialogBoxIndirectA	Creates a modal dialog box from a dialog box template in memory. DialogBoxIndirect does not return control until the specified callback function terminates the modal dialog box by calling the EndDialog function.
DialogBoxIndirectParamA	Creates a modal dialog box from a dialog box template in memory.
DialogBoxIndirectParamW	Creates a modal dialog box from a dialog box template in memory.
DialogBoxIndirectW	Creates a modal dialog box from a dialog box template in memory. DialogBoxIndirect does not return control until the specified callback function terminates the modal dialog box by calling the EndDialog function.
DialogBoxParamA	Creates a modal dialog box from a dialog box template resource.
DialogBoxParamW	Creates a modal dialog box from a dialog box template resource.
DialogBoxW	Creates a modal dialog box from a dialog box template resource. DialogBox does not return control until the specified callback function terminates the modal dialog box by calling the EndDialog function.
DisableProcessWindowsGhosting	Disables the window ghosting feature for the calling GUI process. Window ghosting is a Windows Manager feature that lets the user minimize, move, or close the main window of an application that is not responding.
DispatchMessage	Dispatches a message to a window procedure. It is typically used to dispatch a message retrieved by the GetMessage function.
DispatchMessageA	Dispatches a message to a window procedure. It is typically used to dispatch a message retrieved by the GetMessage function.
DispatchMessageW	Dispatches a message to a window procedure. It is typically used to dispatch a message retrieved by the GetMessage function.
DisplayConfigGetDeviceInfo	The DisplayConfigGetDeviceInfo function retrieves display configuration information about the device.
DisplayConfigSetDeviceInfo	The DisplayConfigSetDeviceInfo function sets the properties of a target.

TITLE	DESCRIPTION
DlGDirListA	Replaces the contents of a list box with the names of the subdirectories and files in a specified directory. You can filter the list of names by specifying a set of file attributes. The list can optionally include mapped drives.
DlGDirListComboBoxA	Replaces the contents of a combo box with the names of the subdirectories and files in a specified directory. You can filter the list of names by specifying a set of file attributes. The list of names can include mapped drive letters.
DlGDirListComboBoxW	Replaces the contents of a combo box with the names of the subdirectories and files in a specified directory. You can filter the list of names by specifying a set of file attributes. The list of names can include mapped drive letters.
DlGDirListW	Replaces the contents of a list box with the names of the subdirectories and files in a specified directory. You can filter the list of names by specifying a set of file attributes. The list can optionally include mapped drives.
DlGDirSelectComboBoxExA	Retrieves the current selection from a combo box filled by using the DlGDirListComboBox function. The selection is interpreted as a drive letter, a file, or a directory name.
DlGDirSelectComboBoxExW	Retrieves the current selection from a combo box filled by using the DlGDirListComboBox function. The selection is interpreted as a drive letter, a file, or a directory name.
DlGDirSelectExA	Retrieves the current selection from a single-selection list box. It assumes that the list box has been filled by the DlGDirList function and that the selection is a drive letter, filename, or directory name.
DlGDirSelectExW	Retrieves the current selection from a single-selection list box. It assumes that the list box has been filled by the DlGDirList function and that the selection is a drive letter, filename, or directory name.
DragDetect	Captures the mouse and tracks its movement until the user releases the left button, presses the ESC key, or moves the mouse outside the drag rectangle around the specified point.
DrawAnimatedRects	Animates the caption of a window to indicate the opening of an icon or the minimizing or maximizing of a window.
DrawCaption	The DrawCaption function draws a window caption.
DrawEdge	The DrawEdge function draws one or more edges of rectangle.
DrawFocusRect	The DrawFocusRect function draws a rectangle in the style used to indicate that the rectangle has the focus.

TITLE	DESCRIPTION
DrawFrameControl	The DrawFrameControl function draws a frame control of the specified type and style.
DrawIcon	Draws an icon or cursor into the specified device context.
DrawIconEx	Draws an icon or cursor into the specified device context, performing the specified raster operations, and stretching or compressing the icon or cursor as specified.
DrawMenuBar	Redraws the menu bar of the specified window. If the menu bar changes after the system has created the window, this function must be called to draw the changed menu bar.
DrawStateA	The DrawState function displays an image and applies a visual effect to indicate a state, such as a disabled or default state.
DrawStateW	The DrawState function displays an image and applies a visual effect to indicate a state, such as a disabled or default state.
DrawText	The DrawText function draws formatted text in the specified rectangle. It formats the text according to the specified method (expanding tabs, justifying characters, breaking lines, and so forth).
DrawTextA	The DrawText function draws formatted text in the specified rectangle. It formats the text according to the specified method (expanding tabs, justifying characters, breaking lines, and so forth).
DrawTextExA	The DrawTextEx function draws formatted text in the specified rectangle.
DrawTextExW	The DrawTextEx function draws formatted text in the specified rectangle.
DrawTextW	The DrawText function draws formatted text in the specified rectangle. It formats the text according to the specified method (expanding tabs, justifying characters, breaking lines, and so forth).
EmptyClipboard	Empties the clipboard and frees handles to data in the clipboard. The function then assigns ownership of the clipboard to the window that currently has the clipboard open.
EnableMenuItem	Enables, disables, or grays the specified menu item.
EnableMouseInPointer	Enables the mouse to act as a pointer input device and send WM_POINTER messages.
EnableNonClientDpiScaling	In high-DPI displays, enables automatic display scaling of the non-client area portions of the specified top-level window. Must be called during the initialization of that window.

TITLE	DESCRIPTION
EnableScrollBar	The EnableScrollBar function enables or disables one or both scroll bar arrows.
EnableWindow	Enables or disables mouse and keyboard input to the specified window or control. When input is disabled, the window does not receive input such as mouse clicks and key presses. When input is enabled, the window receives all input.
EndDeferWindowPos	Simultaneously updates the position and size of one or more windows in a single screen-refreshing cycle.
EndDialog	Destroys a modal dialog box, causing the system to end any processing for the dialog box.
EndMenu	Ends the calling thread's active menu.
EndPoint	The EndPaint function marks the end of painting in the specified window. This function is required for each call to the BeginPaint function, but only after painting is complete.
EndTask	Forcibly closes the specified window.
EnumChildWindows	Enumerates the child windows that belong to the specified parent window by passing the handle to each child window, in turn, to an application-defined callback function.
EnumClipboardFormats	Enumerates the data formats currently available on the clipboard.
EnumDesktopsA	Enumerates all desktops associated with the specified window station of the calling process. The function passes the name of each desktop, in turn, to an application-defined callback function.
EnumDesktopsW	Enumerates all desktops associated with the specified window station of the calling process. The function passes the name of each desktop, in turn, to an application-defined callback function.
EnumDesktopWindows	Enumerates all top-level windows associated with the specified desktop. It passes the handle to each window, in turn, to an application-defined callback function.
EnumDisplayDevicesA	The EnumDisplayDevices function lets you obtain information about the display devices in the current session.
EnumDisplayDevicesW	The EnumDisplayDevices function lets you obtain information about the display devices in the current session.

TITLE	DESCRIPTION
EnumDisplayMonitors	The EnumDisplayMonitors function enumerates display monitors (including invisible pseudo-monitors associated with the mirroring drivers) that intersect a region formed by the intersection of a specified clipping rectangle and the visible region of a device context. EnumDisplayMonitors calls an application-defined MonitorEnumProc callback function once for each monitor that is enumerated. Note that GetSystemMetrics (SM_CMONITORS) counts only the display monitors.
EnumDisplaySettingsA	The EnumDisplaySettings function retrieves information about one of the graphics modes for a display device. To retrieve information for all the graphics modes of a display device, make a series of calls to this function.
EnumDisplaySettingsExA	The EnumDisplaySettingsEx function retrieves information about one of the graphics modes for a display device. To retrieve information for all the graphics modes for a display device, make a series of calls to this function.
EnumDisplaySettingsExW	The EnumDisplaySettingsEx function retrieves information about one of the graphics modes for a display device. To retrieve information for all the graphics modes for a display device, make a series of calls to this function.
EnumDisplaySettingsW	The EnumDisplaySettings function retrieves information about one of the graphics modes for a display device. To retrieve information for all the graphics modes of a display device, make a series of calls to this function.
EnumPropsA	Enumerates all entries in the property list of a window by passing them, one by one, to the specified callback function. EnumProps continues until the last entry is enumerated or the callback function returns FALSE.
EnumPropsExA	Enumerates all entries in the property list of a window by passing them, one by one, to the specified callback function. EnumPropsEx continues until the last entry is enumerated or the callback function returns FALSE.
EnumPropsExW	Enumerates all entries in the property list of a window by passing them, one by one, to the specified callback function. EnumPropsEx continues until the last entry is enumerated or the callback function returns FALSE.
EnumPropsW	Enumerates all entries in the property list of a window by passing them, one by one, to the specified callback function. EnumProps continues until the last entry is enumerated or the callback function returns FALSE.
EnumThreadWindows	Enumerates all nonchild windows associated with a thread by passing the handle to each window, in turn, to an application-defined callback function.

TITLE	DESCRIPTION
EnumWindows	Enumerates all top-level windows on the screen by passing the handle to each window, in turn, to an application-defined callback function. EnumWindows continues until the last top-level window is enumerated or the callback function returns FALSE.
EnumWindowStationsA	Enumerates all window stations in the current session. The function passes the name of each window station, in turn, to an application-defined callback function.
EnumWindowStationsW	Enumerates all window stations in the current session. The function passes the name of each window station, in turn, to an application-defined callback function.
EqualRect	The EqualRect function determines whether the two specified rectangles are equal by comparing the coordinates of their upper-left and lower-right corners.
EvaluateProximityToPolygon	Returns the score of a polygon as the probable touch target (compared to all other polygons that intersect the touch contact area) and an adjusted touch point within the polygon.
EvaluateProximityToRect	Returns the score of a rectangle as the probable touch target, compared to all other rectangles that intersect the touch contact area, and an adjusted touch point within the rectangle.
ExcludeUpdateRgn	The ExcludeUpdateRgn function prevents drawing within invalid areas of a window by excluding an updated region in the window from a clipping region.
ExitWindows	Calls the ExitWindowsEx function to log off the interactive user.
ExitWindowsEx	Logs off the interactive user, shuts down the system, or shuts down and restarts the system.
FillRect	The FillRect function fills a rectangle by using the specified brush. This function includes the left and top borders, but excludes the right and bottom borders of the rectangle.
FindWindowA	Retrieves a handle to the top-level window whose class name and window name match the specified strings. This function does not search child windows. This function does not perform a case-sensitive search.
FindWindowExA	Retrieves a handle to a window whose class name and window name match the specified strings. The function searches child windows, beginning with the one following the specified child window. This function does not perform a case-sensitive search.

TITLE	DESCRIPTION
FindWindowExW	Retrieves a handle to a window whose class name and window name match the specified strings. The function searches child windows, beginning with the one following the specified child window. This function does not perform a case-sensitive search.
FindWindowW	Retrieves a handle to the top-level window whose class name and window name match the specified strings. This function does not search child windows. This function does not perform a case-sensitive search.
FlashWindow	Flashes the specified window one time. It does not change the active state of the window.
FlashWindowEx	Flashes the specified window. It does not change the active state of the window.
FrameRect	The FrameRect function draws a border around the specified rectangle by using the specified brush. The width and height of the border are always one logical unit.
GET_APPCOMMAND_LPARAM	Retrieves the application command from the specified LPARAM value.
GET_DEVICE_LPARAM	Retrieves the input device type from the specified LPARAM value.
GET_FLAGS_LPARAM	Retrieves the state of certain virtual keys from the specified LPARAM value.
GET_KEYSTATE_LPARAM	Retrieves the state of certain virtual keys from the specified LPARAM value.
GET_KEYSTATE_WPARAM	Retrieves the state of certain virtual keys from the specified WPARAM value.
GET_NCHITTEST_WPARAM	Retrieves the hit-test value from the specified WPARAM value.
GET_POINTERID_WPARAM	Retrieves the pointer ID using the specified value.
GET_RAWINPUT_CODE_WPARAM	Retrieves the input code from wParam in WM_INPUT.
GET_WHEEL_DELTA_WPARAM	Retrieves the wheel-delta value from the specified WPARAM value.
GET_XBUTTON_WPARAM	Retrieves the state of certain buttons from the specified WPARAM value.
GetActiveWindow	Retrieves the window handle to the active window attached to the calling thread's message queue.
GetAltTabInfoA	Retrieves status information for the specified window if it is the application-switching (ALT+TAB) window.

TITLE	DESCRIPTION
GetAltTabInfoW	Retrieves status information for the specified window if it is the application-switching (ALT+TAB) window.
GetAncestor	Retrieves the handle to the ancestor of the specified window.
GetAsyncKeyState	Determines whether a key is up or down at the time the function is called, and whether the key was pressed after a previous call to GetAsyncKeyState .
GetAutoRotationState	Retrieves an AR_STATE value containing the state of screen auto-rotation for the system, for example whether auto-rotation is supported, and whether it is enabled by the user.
GetAwarenessFromDpiAwarenessContext	Retrieves the DPI_AWARENESS value from a DPI_AWARENESS_CONTEXT.
GetCapture	Retrieves a handle to the window (if any) that has captured the mouse. Only one window at a time can capture the mouse; this window receives mouse input whether or not the cursor is within its borders.
GetCaretBlinkTime	Retrieves the time required to invert the caret's pixels. The user can set this value.
GetCaretPos	Copies the caret's position to the specified POINT structure.
GetCIMSSM	Retrieves the source of the input message (GetCurrentInputMessageSourceInSendMessage).
GetClassInfoA	Retrieves information about a window class.
GetClassInfoExA	Retrieves information about a window class, including a handle to the small icon associated with the window class. The GetClassInfo function does not retrieve a handle to the small icon.
GetClassInfoExW	Retrieves information about a window class, including a handle to the small icon associated with the window class. The GetClassInfo function does not retrieve a handle to the small icon.
GetClassInfoW	Retrieves information about a window class.
GetClassLongA	Retrieves the specified 32-bit (DWORD) value from the WNDCLASSEX structure associated with the specified window.
GetClassLongPtrA	Retrieves the specified value from the WNDCLASSEX structure associated with the specified window.
GetClassLongPtrW	Retrieves the specified value from the WNDCLASSEX structure associated with the specified window.

TITLE	DESCRIPTION
GetClassLongW	Retrieves the specified 32-bit (DWORD) value from the WNDCLASSEX structure associated with the specified window.
GetClassName	Retrieves the name of the class to which the specified window belongs.
GetClassNameA	Retrieves the name of the class to which the specified window belongs.
GetClassNameW	Retrieves the name of the class to which the specified window belongs.
GetClassWord	Retrieves the 16-bit (WORD) value at the specified offset into the extra class memory for the window class to which the specified window belongs.
GetClientRect	Retrieves the coordinates of a window's client area.
GetClipboardData	Retrieves data from the clipboard in a specified format. The clipboard must have been opened previously.
GetClipboardFormatNameA	Retrieves from the clipboard the name of the specified registered format. The function copies the name to the specified buffer.
GetClipboardFormatNameW	Retrieves from the clipboard the name of the specified registered format. The function copies the name to the specified buffer.
GetClipboardOwner	Retrieves the window handle of the current owner of the clipboard.
GetClipboardSequenceNumber	Retrieves the clipboard sequence number for the current window station.
GetClipboardViewer	Retrieves the handle to the first window in the clipboard viewer chain.
GetClipCursor	Retrieves the screen coordinates of the rectangular area to which the cursor is confined.
GetComboBoxInfo	Retrieves information about the specified combo box.
GetCurrentInputMessageSource	Retrieves the source of the input message.
GetCursor	Retrieves a handle to the current cursor.
GetCursorInfo	Retrieves information about the global cursor.
GetCursorPos	Retrieves the position of the mouse cursor, in screen coordinates.

TITLE	DESCRIPTION
GetDC	The GetDC function retrieves a handle to a device context (DC) for the client area of a specified window or for the entire screen.
GetDCEx	The GetDCEx function retrieves a handle to a device context (DC) for the client area of a specified window or for the entire screen.
GetDesktopWindow	Retrieves a handle to the desktop window. The desktop window covers the entire screen. The desktop window is the area on top of which other windows are painted.
GetDialogBaseUnits	Retrieves the system's dialog base units, which are the average width and height of characters in the system font.
GetDialogControlDpiChangeBehavior	Retrieves and per-monitor DPI scaling behavior overrides of a child window in a dialog.
GetDialogDpiChangeBehavior	Returns the flags that might have been set on a given dialog by an earlier call to SetDialogDpiChangeBehavior.
GetDisplayAutoRotationPreferences	Retrieves the screen auto-rotation preferences for the current process.
GetDisplayAutoRotationPreferencesByProcessId	Retrieves the screen auto-rotation preferences for the process indicated by the dwProcessId parameter.
GetDisplayConfigBufferSizes	The GetDisplayConfigBufferSizes function retrieves the size of the buffers that are required to call the QueryDisplayConfig function.
GetDlgCtrlID	Retrieves the identifier of the specified control.
GetDlgItem	Retrieves a handle to a control in the specified dialog box.
GetDlgItemInt	Translates the text of a specified control in a dialog box into an integer value.
GetDlgItemTextA	Retrieves the title or text associated with a control in a dialog box.
GetDlgItemTextW	Retrieves the title or text associated with a control in a dialog box.
GetDoubleClickTime	Retrieves the current double-click time for the mouse.
GetDpiForSystem	Returns the system DPI.
GetDpiForWindow	Returns the dots per inch (dpi) value for the associated window.

TITLE	DESCRIPTION
GetDpiFromDpiAwarenessContext	Retrieves the DPI from a given DPI_AWARENESS_CONTEXT handle. This enables you to determine the DPI of a thread without needed to examine a window created within that thread.
GetFocus	Retrieves the handle to the window that has the keyboard focus, if the window is attached to the calling thread's message queue.
GetForegroundWindow	Retrieves a handle to the foreground window (the window with which the user is currently working). The system assigns a slightly higher priority to the thread that creates the foreground window than it does to other threads.
GetGestureConfig	Retrieves the configuration for which Windows Touch gesture messages are sent from a window.
GetGestureExtraArgs	Retrieves additional information about a gesture from its GESTUREINFO handle.
GetGestureInfo	Retrieves a GESTUREINFO structure given a handle to the gesture information.
GetGuiResources	Retrieves the count of handles to graphical user interface (GUI) objects in use by the specified process.
GetGUIThreadInfo	Retrieves information about the active window or a specified GUI thread.
GetIconInfo	Retrieves information about the specified icon or cursor.
GetIconInfoExA	Retrieves information about the specified icon or cursor. GetIconInfoEx extends GetIconInfo by using the newer ICONINFOEX structure.
GetIconInfoExW	Retrieves information about the specified icon or cursor. GetIconInfoEx extends GetIconInfo by using the newer ICONINFOEX structure.
GetInputState	Determines whether there are mouse-button or keyboard messages in the calling thread's message queue.
GetKBCodePage	Retrieves the current code page.
GetKeyboardLayout	Retrieves the active input locale identifier (formerly called the keyboard layout).
GetKeyboardLayoutList	Retrieves the input locale identifiers (formerly called keyboard layout handles) corresponding to the current set of input locales in the system. The function copies the identifiers to the specified buffer.
GetKeyboardLayoutNameA	Retrieves the name of the active input locale identifier (formerly called the keyboard layout) for the system.

TITLE	DESCRIPTION
GetKeyboardLayoutNameW	Retrieves the name of the active input locale identifier (formerly called the keyboard layout) for the system.
GetKeyboardState	Copies the status of the 256 virtual keys to the specified buffer.
GetKeyboardType	Retrieves information about the current keyboard.
GetKeyNameTextA	Retrieves a string that represents the name of a key.
GetKeyNameTextW	Retrieves a string that represents the name of a key.
GetKeyState	Retrieves the status of the specified virtual key. The status specifies whether the key is up, down, or toggled (on, off, alternating each time the key is pressed).
GetLastActivePopup	Determines which pop-up window owned by the specified window was most recently active.
GetLastInputInfo	Retrieves the time of the last input event.
GetLayeredWindowAttributes	Retrieves the opacity and transparency color key of a layered window.
GetListBoxInfo	Retrieves the number of items per column in a specified list box.
GetMenu	Retrieves a handle to the menu assigned to the specified window.
GetMenuBarInfo	Retrieves information about the specified menu bar.
GetMenuCheckMarkDimensions	Retrieves the dimensions of the default check-mark bitmap.
GetMenuContextHelpId	Retrieves the Help context identifier associated with the specified menu.
GetMenuDefaultItem	Determines the default menu item on the specified menu.
GetMenuInfo	Retrieves information about a specified menu.
GetMenuItemCount	Determines the number of items in the specified menu.
GetMenuItemID	Retrieves the menu item identifier of a menu item located at the specified position in a menu.
GetMenuItemInfoA	Retrieves information about a menu item.
GetMenuItemInfoW	Retrieves information about a menu item.
GetMenuItemRect	Retrieves the bounding rectangle for the specified menu item.

TITLE	DESCRIPTION
GetMenuState	Retrieves the menu flags associated with the specified menu item.
GetMenuStringA	Copies the text string of the specified menu item into the specified buffer.
GetMenuStringW	Copies the text string of the specified menu item into the specified buffer.
GetMessage	Retrieves a message from the calling thread's message queue. The function dispatches incoming sent messages until a posted message is available for retrieval.
GetMessageA	Retrieves a message from the calling thread's message queue. The function dispatches incoming sent messages until a posted message is available for retrieval.
GetMessageExtraInfo	Retrieves the extra message information for the current thread. Extra message information is an application- or driver-defined value associated with the current thread's message queue.
GetMessagePos	Retrieves the cursor position for the last message retrieved by the GetMessage function.
GetMessageTime	Retrieves the message time for the last message retrieved by the GetMessage function.
GetMessageW	Retrieves a message from the calling thread's message queue. The function dispatches incoming sent messages until a posted message is available for retrieval.
GetMonitorInfoA	The GetMonitorInfo function retrieves information about a display monitor.
GetMonitorInfoW	The GetMonitorInfo function retrieves information about a display monitor.
GetMouseMovePointsEx	Retrieves a history of up to 64 previous coordinates of the mouse or pen.
GetNextDlgGroupItem	Retrieves a handle to the first control in a group of controls that precedes (or follows) the specified control in a dialog box.
GetNextDlgTabItem	Retrieves a handle to the first control that has the WS_TABSTOP style that precedes (or follows) the specified control.
GetNextWindow	Retrieves a handle to the next or previous window in the Z-Order. The next window is below the specified window; the previous window is above.
GetOpenClipboardWindow	Retrieves the handle to the window that currently has the clipboard open.

TITLE	DESCRIPTION
GetParent	Retrieves a handle to the specified window's parent or owner.
GetPhysicalCursorPos	Retrieves the position of the cursor in physical coordinates.
GetPointerCursorId	Retrieves the cursor identifier associated with the specified pointer.
GetPointerDevice	Gets information about the pointer device.
GetPointerDeviceCursors	Gets the cursor IDs that are mapped to the cursors associated with a pointer device.
GetPointerDeviceProperties	Gets device properties that aren't included in the POINTER_DEVICE_INFO structure.
GetPointerDeviceRects	Gets the x and y range for the pointer device (in himetric) and the x and y range (current resolution) for the display that the pointer device is mapped to.
GetPointerDevices	Gets information about the pointer devices attached to the system.
GetPointerFrameInfo	Gets the entire frame of information for the specified pointers associated with the current message.
GetPointerFrameInfoHistory	Gets the entire frame of information (including coalesced input frames) for the specified pointers associated with the current message.
GetPointerFramePenInfo	Gets the entire frame of pen-based information for the specified pointers (of type PT_PEN) associated with the current message.
GetPointerFramePenInfoHistory	Gets the entire frame of pen-based information (including coalesced input frames) for the specified pointers (of type PT_PEN) associated with the current message.
GetPointerFrameTouchInfo	Gets the entire frame of touch-based information for the specified pointers (of type PT_TOUCH) associated with the current message.
GetPointerFrameTouchInfoHistory	Gets the entire frame of touch-based information (including coalesced input frames) for the specified pointers (of type PT_TOUCH) associated with the current message.
GetPointerInfo	Gets the information for the specified pointer associated with the current message.
GetPointerInfoHistory	Gets the information associated with the individual inputs, if any, that were coalesced into the current message for the specified pointer.

TITLE	DESCRIPTION
GetPointerInputTransform	Gets one or more transforms for the pointer information coordinates associated with the current message.
GetPointerPenInfo	Gets the pen-based information for the specified pointer (of type PT_PEN) associated with the current message.
GetPointerPenInfoHistory	Gets the pen-based information associated with the individual inputs, if any, that were coalesced into the current message for the specified pointer (of type PT_PEN).
GetPointerTouchInfo	Gets the touch-based information for the specified pointer (of type PT_TOUCH) associated with the current message.
GetPointerTouchInfoHistory	Gets the touch-based information associated with the individual inputs, if any, that were coalesced into the current message for the specified pointer (of type PT_TOUCH).
GetPointerType	Retrieves the pointer type for a specified pointer.
GetPriorityClipboardFormat	Retrieves the first available clipboard format in the specified list.
GetProcessDefaultLayout	Retrieves the default layout that is used when windows are created with no parent or owner.
GetProcessWindowStation	Retrieves a handle to the current window station for the calling process.
GetPropA	Retrieves a data handle from the property list of the specified window. The character string identifies the handle to be retrieved. The string and handle must have been added to the property list by a previous call to the SetProp function.
GetPropW	Retrieves a data handle from the property list of the specified window. The character string identifies the handle to be retrieved. The string and handle must have been added to the property list by a previous call to the SetProp function.
GetQueueStatus	Retrieves the type of messages found in the calling thread's message queue.
GetRawInputBuffer	Performs a buffered read of the raw input data.
GetRawInputData	Retrieves the raw input from the specified device.
GetRawInputDeviceInfoA	Retrieves information about the raw input device.
GetRawInputDeviceInfoW	Retrieves information about the raw input device.
GetRawInputDeviceList	Enumerates the raw input devices attached to the system.
GetRawPointerDeviceData	Gets the raw input data from the pointer device.

TITLE	DESCRIPTION
GetRegisteredRawInputDevices	Retrieves the information about the raw input devices for the current application.
GetScrollBarInfo	The GetScrollBarInfo function retrieves information about the specified scroll bar.
GetScrollInfo	The GetScrollInfo function retrieves the parameters of a scroll bar, including the minimum and maximum scrolling positions, the page size, and the position of the scroll box (thumb).
GetScrollPos	The GetScrollPos function retrieves the current position of the scroll box (thumb) in the specified scroll bar.
GetScrollRange	The GetScrollRange function retrieves the current minimum and maximum scroll box (thumb) positions for the specified scroll bar.
GetShellWindow	Retrieves a handle to the Shell's desktop window.
GetSubMenu	Retrieves a handle to the drop-down menu or submenu activated by the specified menu item.
GetSysColor	Retrieves the current color of the specified display element.
GetSysColorBrush	The GetSysColorBrush function retrieves a handle identifying a logical brush that corresponds to the specified color index.
GetSystemDpiForProcess	Retrieves the system DPI associated with a given process. This is useful for avoiding compatibility issues that arise from sharing DPI-sensitive information between multiple system-aware processes with different system DPI values.
GetSystemMenu	Enables the application to access the window menu (also known as the system menu or the control menu) for copying and modifying.
GetSystemMetrics	Retrieves the specified system metric or system configuration setting.
GetSystemMetricsForDpi	Retrieves the specified system metric or system configuration setting taking into account a provided DPI.
GetTabbedTextExtentA	The GetTabbedTextExtent function computes the width and height of a character string.
GetTabbedTextExtentW	The GetTabbedTextExtent function computes the width and height of a character string.
GetThreadDesktop	Retrieves a handle to the desktop assigned to the specified thread.
GetThreadDpiAwarenessContext	Gets the DPI_AWARENESS_CONTEXT for the current thread.

TITLE	DESCRIPTION
GetThreadDpiHostingBehavior	Retrieves the DPI_HOSTING_BEHAVIOR from the current thread.
GetTitleBarInfo	Retrieves information about the specified title bar.
GetTopWindow	Examines the Z order of the child windows associated with the specified parent window and retrieves a handle to the child window at the top of the Z order.
GetTouchInputInfo	Retrieves detailed information about touch inputs associated with a particular touch input handle.
GetUnpredictedMessagePos	Gets pointer data before it has gone through touch prediction processing.
GetUpdatedClipboardFormats	Retrieves the currently supported clipboard formats.
GetUpdateRect	The GetUpdateRect function retrieves the coordinates of the smallest rectangle that completely encloses the update region of the specified window.
GetUpdateRgn	The GetUpdateRgn function retrieves the update region of a window by copying it into the specified region. The coordinates of the update region are relative to the upper-left corner of the window (that is, they are client coordinates).
GetUserObjectInformationA	Retrieves information about the specified window station or desktop object.
GetUserObjectInformationW	Retrieves information about the specified window station or desktop object.
GetUserObjectSecurity	Retrieves security information for the specified user object.
GetWindow	Retrieves a handle to a window that has the specified relationship (Z-Order or owner) to the specified window.
GetWindowContextHelpId	Retrieves the Help context identifier, if any, associated with the specified window.
GetWindowDC	The GetWindowDC function retrieves the device context (DC) for the entire window, including title bar, menus, and scroll bars.
GetWindowDisplayAffinity	Retrieves the current display affinity setting, from any process, for a given window.
GetWindowDpiAwarenessContext	Returns the DPI_AWARENESS_CONTEXT associated with a window.
GetWindowDpiHostingBehavior	Returns the DPI_HOSTING_BEHAVIOR of the specified window.

TITLE	DESCRIPTION
GetWindowFeedbackSetting	Retrieves the feedback configuration for a window.
GetWindowInfo	Retrieves information about the specified window.
GetWindowLongA	Retrieves information about the specified window.
GetWindowLongPtrA	Retrieves information about the specified window. The function also retrieves the value at a specified offset into the extra window memory.
GetWindowLongPtrW	Retrieves information about the specified window. The function also retrieves the value at a specified offset into the extra window memory.
GetWindowLongW	Retrieves information about the specified window.
GetWindowModuleFileNameA	Retrieves the full path and file name of the module associated with the specified window handle.
GetWindowModuleFileNameW	Retrieves the full path and file name of the module associated with the specified window handle.
GetWindowPlacement	Retrieves the show state and the restored, minimized, and maximized positions of the specified window.
GetWindowRect	Retrieves the dimensions of the bounding rectangle of the specified window. The dimensions are given in screen coordinates that are relative to the upper-left corner of the screen.
GetWindowRgn	The GetWindowRgn function obtains a copy of the window region of a window.
GetWindowRgnBox	The GetWindowRgnBox function retrieves the dimensions of the tightest bounding rectangle for the window region of a window.
GetWindowTextA	Copies the text of the specified window's title bar (if it has one) into a buffer. If the specified window is a control, the text of the control is copied. However, GetWindowText cannot retrieve the text of a control in another application.
GetWindowTextLengthA	Retrieves the length, in characters, of the specified window's title bar text (if the window has a title bar).
GetWindowTextLengthW	Retrieves the length, in characters, of the specified window's title bar text (if the window has a title bar).
GetWindowTextW	Copies the text of the specified window's title bar (if it has one) into a buffer. If the specified window is a control, the text of the control is copied. However, GetWindowText cannot retrieve the text of a control in another application.

TITLE	DESCRIPTION
GetWindowThreadProcessId	Retrieves the identifier of the thread that created the specified window and, optionally, the identifier of the process that created the window.
GID_ROTATE_ANGLE_FROM_ARGUMENT	The GID_ROTATE_ANGLE_FROM_ARGUMENT macro is used to interpret the GID_ROTATE ullArgument value when receiving the value in the WM_GESTURE structure.
GID_ROTATE_ANGLE_TO_ARGUMENT	Converts a radian value to an argument for rotation gesture messages.
GrayStringA	The GrayString function draws gray text at the specified location.
GrayStringW	The GrayString function draws gray text at the specified location.
HAS_POINTER_CONFIDENCE_WPARAM	Checks whether the specified pointer message is considered intentional rather than accidental.
HideCaret	Removes the caret from the screen. Hiding a caret does not destroy its current shape or invalidate the insertion point.
HiliteMenuItem	Adds or removes highlighting from an item in a menu bar.
InflateRect	The InflateRect function increases or decreases the width and height of the specified rectangle.
InitializeTouchInjection	Configures the touch injection context for the calling application and initializes the maximum number of simultaneous contacts that the app can inject.
InjectSyntheticPointerInput	Simulates pointer input (pen or touch).
InjectTouchInput	Simulates touch input.
InSendMessage	Determines whether the current window procedure is processing a message that was sent from another thread (in the same process or a different process) by a call to the SendMessage function.
InSendMessageEx	Determines whether the current window procedure is processing a message that was sent from another thread (in the same process or a different process).
InsertMenuA	Inserts a new menu item into a menu, moving other items down the menu.
InsertMenuItemA	Inserts a new menu item at the specified position in a menu.
InsertMenuItemW	Inserts a new menu item at the specified position in a menu.
InsertMenuW	Inserts a new menu item into a menu, moving other items down the menu.

TITLE	DESCRIPTION
InternalGetWindowText	Copies the text of the specified window's title bar (if it has one) into a buffer.
IntersectRect	The IntersectRect function calculates the intersection of two source rectangles and places the coordinates of the intersection rectangle into the destination rectangle.
InvalidateRect	The InvalidateRect function adds a rectangle to the specified window's update region. The update region represents the portion of the window's client area that must be redrawn.
InvalidateRgn	The InvalidateRgn function invalidates the client area within the specified region by adding it to the current update region of a window.
InvertRect	The InvertRect function inverts a rectangle in a window by performing a logical NOT operation on the color values for each pixel in the rectangle's interior.
IS_INTRESOURCE	Determines whether a value is an integer identifier for a resource.
IS_POINTER_CANCELED_WPARAM	Checks whether the specified pointer input ended abruptly, or was invalid, indicating the interaction was not completed.
IS_POINTER_FIFTHBUTTON_WPARAM	Checks whether the specified pointer took fifth action.
IS_POINTER_FIRSTBUTTON_WPARAM	Checks whether the specified pointer took first action.
IS_POINTER_FLAG_SET_WPARAM	Checks whether a pointer macro sets the specified flag.
IS_POINTER_FOURTHBUTTON_WPARAM	Checks whether the specified pointer took fourth action.
IS_POINTER_INCONTACT_WPARAM	Checks whether the specified pointer is in contact.
IS_POINTER_INRANGE_WPARAM	Checks whether the specified pointer is in range.
IS_POINTER_NEW_WPARAM	Checks whether the specified pointer is a new pointer.
IS_POINTER_SECONDBUTTON_WPARAM	Checks whether the specified pointer took second action.
IS_POINTER_THIRDBUTTON_WPARAM	Checks whether the specified pointer took third action.
IsCharAlphaA	Determines whether a character is an alphabetical character. This determination is based on the semantics of the language selected by the user during setup or through Control Panel.
IsCharAlphaNumericA	Determines whether a character is either an alphabetical or a numeric character. This determination is based on the semantics of the language selected by the user during setup or through Control Panel.

TITLE	DESCRIPTION
IsCharAlphaNumericW	Determines whether a character is either an alphabetical or a numeric character. This determination is based on the semantics of the language selected by the user during setup or through Control Panel.
IsCharAlphaW	Determines whether a character is an alphabetical character. This determination is based on the semantics of the language selected by the user during setup or through Control Panel.
IsCharLowerA	Determines whether a character is lowercase. This determination is based on the semantics of the language selected by the user during setup or through Control Panel.
IsCharLowerW	
IsCharUpperA	Determines whether a character is uppercase. This determination is based on the semantics of the language selected by the user during setup or through Control Panel.
IsCharUpperW	Determines whether a character is uppercase. This determination is based on the semantics of the language selected by the user during setup or through Control Panel.
IsChild	Determines whether a window is a child window or descendant window of a specified parent window.
IsClipboardFormatAvailable	Determines whether the clipboard contains data in the specified format.
IsDialogMessageA	Determines whether a message is intended for the specified dialog box and, if it is, processes the message.
IsDialogMessageW	Determines whether a message is intended for the specified dialog box and, if it is, processes the message.
IsDlgButtonChecked	The IsDlgButtonChecked function determines whether a button control is checked or whether a three-state button control is checked, unchecked, or indeterminate.
IsGUIThread	Determines whether the calling thread is already a GUI thread. It can also optionally convert the thread to a GUI thread.
IsHungAppWindow	Determines whether the system considers that a specified application is not responding.
IsIconic	Determines whether the specified window is minimized (iconic).
IsImmersiveProcess	Determines whether the process belongs to a Windows Store app.
IsMenu	Determines whether a handle is a menu handle.

TITLE	DESCRIPTION
IsMouseInPointerEnabled	Indicates whether EnableMouseInPointer is set for the mouse to act as a pointer input device and send WM_POINTER messages.
IsProcessDPIAware	IsProcessDPIAware may be altered or unavailable. Instead, use GetProcessDPIAwareness.
IsRectEmpty	The IsRectEmpty function determines whether the specified rectangle is empty.
IsTouchWindow	Checks whether a specified window is touch-capable and, optionally, retrieves the modifier flags set for the window's touch capability.
IsValidDpiAwarenessContext	Determines if a specified DPI_AWARENESS_CONTEXT is valid and supported by the current system.
IsWindow	Determines whether the specified window handle identifies an existing window.
IsWindowEnabled	Determines whether the specified window is enabled for mouse and keyboard input.
IsWindowUnicode	Determines whether the specified window is a native Unicode window.
IsWindowVisible	Determines the visibility state of the specified window.
IsWinEventHookInstalled	Determines whether there is an installed WinEvent hook that might be notified of a specified event.
IsWow64Message	Determines whether the last message read from the current thread's queue originated from a WOW64 process.
IsZoomed	Determines whether a window is maximized.
keybd_event	Synthesizes a keystroke.
KillTimer	Destroys the specified timer.
LoadAcceleratorsA	Loads the specified accelerator table.
LoadAcceleratorsW	Loads the specified accelerator table.
LoadBitmapA	The LoadBitmap function loads the specified bitmap resource from a module's executable file.
LoadBitmapW	The LoadBitmap function loads the specified bitmap resource from a module's executable file.
LoadCursorA	Loads the specified cursor resource from the executable (.EXE) file associated with an application instance.

TITLE	DESCRIPTION
LoadCursorFromFileA	Creates a cursor based on data contained in a file.
LoadCursorFromFileW	Creates a cursor based on data contained in a file.
LoadCursorW	Loads the specified cursor resource from the executable (.EXE) file associated with an application instance.
LoadIconA	Loads the specified icon resource from the executable (.exe) file associated with an application instance.
LoadIconW	Loads the specified icon resource from the executable (.exe) file associated with an application instance.
LoadImageA	Loads an icon, cursor, animated cursor, or bitmap.
LoadImageW	Loads an icon, cursor, animated cursor, or bitmap.
LoadKeyboardLayoutA	Loads a new input locale identifier (formerly called the keyboard layout) into the system.
LoadKeyboardLayoutW	Loads a new input locale identifier (formerly called the keyboard layout) into the system.
LoadMenuA	Loads the specified menu resource from the executable (.exe) file associated with an application instance.
LoadMenuIndirectA	Loads the specified menu template in memory.
LoadMenuIndirectW	Loads the specified menu template in memory.
LoadMenuW	Loads the specified menu resource from the executable (.exe) file associated with an application instance.
LoadStringA	Loads a string resource from the executable file associated with a specified module, copies the string into a buffer, and appends a terminating null character.
LoadStringW	Loads a string resource from the executable file associated with a specified module, copies the string into a buffer, and appends a terminating null character.
LockSetForegroundWindow	The foreground process can call the LockSetForegroundWindow function to disable calls to the SetForegroundWindow function.
LockWindowUpdate	The LockWindowUpdate function disables or enables drawing in the specified window. Only one window can be locked at a time.
LockWorkStation	Locks the workstation's display.
LogicalToPhysicalPoint	Converts the logical coordinates of a point in a window to physical coordinates.

TITLE	DESCRIPTION
LogicalToPhysicalPointForPerMonitorDPI	Converts a point in a window from logical coordinates into physical coordinates, regardless of the dots per inch (dpi) awareness of the caller.
LookupIconIdFromDirectory	Searches through icon or cursor data for the icon or cursor that best fits the current display device.
LookupIconIdFromDirectoryEx	Searches through icon or cursor data for the icon or cursor that best fits the current display device.
MAKEINTRESOURCEA	Converts an integer value to a resource type compatible with the resource-management functions. This macro is used in place of a string containing the name of the resource.
MAKEINTRESOURCEW	Converts an integer value to a resource type compatible with the resource-management functions. This macro is used in place of a string containing the name of the resource.
MAKELPARAM	Creates a value for use as an lParam parameter in a message. The macro concatenates the specified values.
MAKELRESULT	Creates a value for use as a return value from a window procedure. The macro concatenates the specified values.
MAKEWPARAM	Creates a value for use as a wParam parameter in a message. The macro concatenates the specified values.
MapDialogRect	Converts the specified dialog box units to screen units (pixels).
MapVirtualKeyA	Translates (maps) a virtual-key code into a scan code or character value, or translates a scan code into a virtual-key code.
MapVirtualKeyExA	Translates (maps) a virtual-key code into a scan code or character value, or translates a scan code into a virtual-key code. The function translates the codes using the input language and an input locale identifier.
MapVirtualKeyExW	Translates (maps) a virtual-key code into a scan code or character value, or translates a scan code into a virtual-key code. The function translates the codes using the input language and an input locale identifier.
MapVirtualKeyW	Translates (maps) a virtual-key code into a scan code or character value, or translates a scan code into a virtual-key code.
MapWindowPoints	The MapWindowPoints function converts (maps) a set of points from a coordinate space relative to one window to a coordinate space relative to another window.
MenuItemFromPoint	Determines which menu item, if any, is at the specified location.

TITLE	DESCRIPTION
MessageBeep	Plays a waveform sound. The waveform sound for each sound type is identified by an entry in the registry.
MessageBox	Displays a modal dialog box that contains a system icon, a set of buttons, and a brief application-specific message, such as status or error information. The message box returns an integer value that indicates which button the user clicked.
MessageBoxA	Displays a modal dialog box that contains a system icon, a set of buttons, and a brief application-specific message, such as status or error information. The message box returns an integer value that indicates which button the user clicked.
MessageBoxExA	Creates, displays, and operates a message box.
MessageBoxExW	Creates, displays, and operates a message box.
MessageBoxIndirectA	Creates, displays, and operates a message box. The message box contains application-defined message text and title, any icon, and any combination of predefined push buttons.
MessageBoxIndirectW	Creates, displays, and operates a message box. The message box contains application-defined message text and title, any icon, and any combination of predefined push buttons.
MessageBoxW	Displays a modal dialog box that contains a system icon, a set of buttons, and a brief application-specific message, such as status or error information. The message box returns an integer value that indicates which button the user clicked.
ModifyMenuA	Changes an existing menu item.
ModifyMenuW	Changes an existing menu item.
MonitorFromPoint	The MonitorFromPoint function retrieves a handle to the display monitor that contains a specified point.
MonitorFromRect	The MonitorFromRect function retrieves a handle to the display monitor that has the largest area of intersection with a specified rectangle.
MonitorFromWindow	The MonitorFromWindow function retrieves a handle to the display monitor that has the largest area of intersection with the bounding rectangle of a specified window.
mouse_event	The mouse_event function synthesizes mouse motion and button clicks.
MoveWindow	Changes the position and dimensions of the specified window.
MsgWaitForMultipleObjects	Waits until one or all of the specified objects are in the signaled state or the time-out interval elapses. The objects can include input event objects.

TITLE	DESCRIPTION
MsgWaitForMultipleObjectsEx	Waits until one or all of the specified objects are in the signaled state, an I/O completion routine or asynchronous procedure call (APC) is queued to the thread, or the time-out interval elapses. The array of objects can include input event objects.
NEXTRAWINPUTBLOCK	Retrieves the location of the next structure in an array of RAWINPUT structures.
NotifyWinEvent	Signals the system that a predefined event occurred. If any client applications have registered a hook function for the event, the system calls the client's hook function.
OemKeyScan	Maps OEMASCII codes 0 through 0x0FF into the OEM scan codes and shift states. The function provides information that allows a program to send OEM text to another program by simulating keyboard input.
OemToCharA	Translates a string from the OEM-defined character set into either an ANSI or a wide-character string. Warning Do not use.
OemToCharBuffA	Translates a specified number of characters in a string from the OEM-defined character set into either an ANSI or a wide-character string.
OemToCharBuffW	Translates a specified number of characters in a string from the OEM-defined character set into either an ANSI or a wide-character string.
OemToCharW	Translates a string from the OEM-defined character set into either an ANSI or a wide-character string. Warning Do not use.
OffsetRect	The OffsetRect function moves the specified rectangle by the specified offsets.
OpenClipboard	Opens the clipboard for examination and prevents other applications from modifying the clipboard content.
OpenDesktopA	Opens the specified desktop object.
OpenDesktopW	Opens the specified desktop object.
OpenIcon	Restores a minimized (iconic) window to its previous size and position; it then activates the window.
OpenInputDesktop	Opens the desktop that receives user input.
OpenWindowStationA	Opens the specified window station.
OpenWindowStationW	Opens the specified window station.

TITLE	DESCRIPTION
PackTouchHitTestingProximityEvaluation	Returns the proximity evaluation score and the adjusted touch-point coordinates as a packed value for the WM_TOUCHHITTESTING callback.
PaintDesktop	The PaintDesktop function fills the clipping region in the specified device context with the desktop pattern or wallpaper. The function is provided primarily for shell desktops.
PeekMessageA	Dispatches incoming sent messages, checks the thread message queue for a posted message, and retrieves the message (if any exist).
PeekMessageW	Dispatches incoming sent messages, checks the thread message queue for a posted message, and retrieves the message (if any exist).
PhysicalToLogicalPoint	Converts the physical coordinates of a point in a window to logical coordinates.
PhysicalToLogicalPointForPerMonitorDPI	Converts a point in a window from physical coordinates into logical coordinates, regardless of the dots per inch (dpi) awareness of the caller.
POINTSTOPOINT	The POINTSTOPOINT macro copies the contents of a POINTS structure into a POINT structure.
POINTTOPOINTS	The POINTTOPOINTS macro converts a POINT structure to a POINTS structure.
PostMessageA	Places (posts) a message in the message queue associated with the thread that created the specified window and returns without waiting for the thread to process the message.
PostMessageW	Places (posts) a message in the message queue associated with the thread that created the specified window and returns without waiting for the thread to process the message.
PostQuitMessage	Indicates to the system that a thread has made a request to terminate (quit). It is typically used in response to a WM_DESTROY message.
PostThreadMessageA	Posts a message to the message queue of the specified thread. It returns without waiting for the thread to process the message.
PostThreadMessageW	Posts a message to the message queue of the specified thread. It returns without waiting for the thread to process the message.
PrintWindow	The PrintWindow function copies a visual window into the specified device context (DC), typically a printer DC.

TITLE	DESCRIPTION
PrivateExtractIconsA	Creates an array of handles to icons that are extracted from a specified file.
PrivateExtractIconsW	Creates an array of handles to icons that are extracted from a specified file.
PtInRect	The PtInRect function determines whether the specified point lies within the specified rectangle.
QueryDisplayConfig	The QueryDisplayConfig function retrieves information about all possible display paths for all display devices, or views, in the current setting.
RealChildWindowFromPoint	Retrieves a handle to the child window at the specified point. The search is restricted to immediate child windows; grandchildren and deeper descendant windows are not searched.
RealGetWindowClassW	Retrieves a string that specifies the window type.
RedrawWindow	The RedrawWindow function updates the specified rectangle or region in a window's client area.
RegisterClassA	Registers a window class for subsequent use in calls to the CreateWindow or CreateWindowEx function.
RegisterClassExA	Registers a window class for subsequent use in calls to the CreateWindow or CreateWindowEx function.
RegisterClassExW	Registers a window class for subsequent use in calls to the CreateWindow or CreateWindowEx function.
RegisterClassW	Registers a window class for subsequent use in calls to the CreateWindow or CreateWindowEx function.
RegisterClipboardFormatA	Registers a new clipboard format. This format can then be used as a valid clipboard format.
RegisterClipboardFormatW	Registers a new clipboard format. This format can then be used as a valid clipboard format.
RegisterDeviceNotificationA	Registers the device or type of device for which a window will receive notifications.
RegisterDeviceNotificationW	Registers the device or type of device for which a window will receive notifications.
RegisterHotKey	Defines a system-wide hot key.
RegisterPointerDeviceNotifications	Registers a window to process the WM_POINTERDEVICECHANGE, WM_POINTERDEVICEINRANGE, and WM_POINTERDEVICEOUTOFRANGE pointer device notifications.

TITLE	DESCRIPTION
RegisterPointerInputTarget	Allows the caller to register a target window to which all pointer input of the specified type is redirected.
RegisterPointerInputTargetEx	RegisterPointerInputTargetEx may be altered or unavailable. Instead, use RegisterPointerInputTarget.
RegisterPowerSettingNotification	Registers the application to receive power setting notifications for the specific power setting event.
RegisterRawInputDevices	Registers the devices that supply the raw input data.
RegisterShellHookWindow	Registers a specified Shell window to receive certain messages for events or notifications that are useful to Shell applications.
RegisterSuspendResumeNotification	Registers to receive notification when the system is suspended or resumed. Similar to PowerRegisterSuspendResumeNotification, but operates in user mode and can take a window handle.
RegisterTouchHitTestingWindow	Registers a window to process the WM_TOUCHHITTESTING notification.
RegisterTouchWindow	Registers a window as being touch-capable.
RegisterWindowMessageA	Defines a new window message that is guaranteed to be unique throughout the system. The message value can be used when sending or posting messages.
RegisterWindowMessageW	Defines a new window message that is guaranteed to be unique throughout the system. The message value can be used when sending or posting messages.
ReleaseCapture	Releases the mouse capture from a window in the current thread and restores normal mouse input processing.
ReleaseDC	The ReleaseDC function releases a device context (DC), freeing it for use by other applications. The effect of the ReleaseDC function depends on the type of DC. It frees only common and window DCs. It has no effect on class or private DCs.
RemoveClipboardFormatListener	Removes the given window from the system-maintained clipboard format listener list.
RemoveMenu	Deletes a menu item or detaches a submenu from the specified menu.
RemovePropA	Removes an entry from the property list of the specified window. The specified character string identifies the entry to be removed.
RemovePropW	Removes an entry from the property list of the specified window. The specified character string identifies the entry to be removed.

TITLE	DESCRIPTION
ReplyMessage	Replies to a message sent from another thread by the SendMessage function.
ScreenToClient	The ScreenToClient function converts the screen coordinates of a specified point on the screen to client-area coordinates.
ScrollDC	The ScrollDC function scrolls a rectangle of bits horizontally and vertically.
ScrollWindow	The ScrollWindow function scrolls the contents of the specified window's client area.
ScrollWindowEx	The ScrollWindowEx function scrolls the contents of the specified window's client area.
SendDlgItemMessageA	Sends a message to the specified control in a dialog box.
SendDlgItemMessageW	Sends a message to the specified control in a dialog box.
SendInput	Synthesizes keystrokes, mouse motions, and button clicks.
SendMessage	Sends the specified message to a window or windows. The SendMessage function calls the window procedure for the specified window and does not return until the window procedure has processed the message.
SendMessageA	Sends the specified message to a window or windows. The SendMessage function calls the window procedure for the specified window and does not return until the window procedure has processed the message.
SendMessageCallbackA	Sends the specified message to a window or windows.
SendMessageCallbackW	Sends the specified message to a window or windows.
SendMessageTimeoutA	Sends the specified message to one or more windows.
SendMessageTimeoutW	Sends the specified message to one or more windows.
SendMessageW	Sends the specified message to a window or windows. The SendMessage function calls the window procedure for the specified window and does not return until the window procedure has processed the message.
SendNotifyMessageA	Sends the specified message to a window or windows.
SendNotifyMessageW	Sends the specified message to a window or windows.

TITLE	DESCRIPTION
SetActiveWindow	Activates a window. The window must be attached to the calling thread's message queue.
SetCapture	Sets the mouse capture to the specified window belonging to the current thread.
SetCaretBlinkTime	Sets the caret blink time to the specified number of milliseconds. The blink time is the elapsed time, in milliseconds, required to invert the caret's pixels.
SetCaretPos	Moves the caret to the specified coordinates. If the window that owns the caret was created with the CS_OWNDC class style, then the specified coordinates are subject to the mapping mode of the device context associated with that window.
SetClassLongA	Replaces the specified 32-bit (long) value at the specified offset into the extra class memory or the WNDCLASSEX structure for the class to which the specified window belongs.
SetClassLongPtrA	Replaces the specified value at the specified offset in the extra class memory or the WNDCLASSEX structure for the class to which the specified window belongs.
SetClassLongPtrW	Replaces the specified value at the specified offset in the extra class memory or the WNDCLASSEX structure for the class to which the specified window belongs.
SetClassLongW	Replaces the specified 32-bit (long) value at the specified offset into the extra class memory or the WNDCLASSEX structure for the class to which the specified window belongs.
SetClassWord	Replaces the 16-bit (WORD) value at the specified offset into the extra class memory for the window class to which the specified window belongs.
SetClipboardData	Places data on the clipboard in a specified clipboard format.
SetClipboardViewer	Adds the specified window to the chain of clipboard viewers. Clipboard viewer windows receive a WM_DRAWCLIPBOARD message whenever the content of the clipboard changes. This function is used for backward compatibility with earlier versions of Windows.
SetCoalescableTimer	Creates a timer with the specified time-out value and coalescing tolerance delay.
SetCursor	Sets the cursor shape.
SetCursorPos	Moves the cursor to the specified screen coordinates.
SetDialogControlDpiChangeBehavior	Overrides the default per-monitor DPI scaling behavior of a child window in a dialog.

TITLE	DESCRIPTION
SetDialogDpiChangeBehavior	Dialogs in Per-Monitor v2 contexts are automatically DPI scaled. This method lets you customize their DPI change behavior.
SetDisplayAutoRotationPreferences	Sets the screen auto-rotation preferences for the current process.
SetDisplayConfig	The SetDisplayConfig function modifies the display topology, source, and target modes by exclusively enabling the specified paths in the current session.
SetDlgItemInt	Sets the text of a control in a dialog box to the string representation of a specified integer value.
SetDlgItemTextA	Sets the title or text of a control in a dialog box.
SetDlgItemTextW	Sets the title or text of a control in a dialog box.
SetDoubleClickTime	Sets the double-click time for the mouse.
SetFocus	Sets the keyboard focus to the specified window. The window must be attached to the calling thread's message queue.
SetForegroundWindow	Brings the thread that created the specified window into the foreground and activates the window.
SetGestureConfig	Configures the messages that are sent from a window for Windows Touch gestures.
SetKeyboardState	Copies an array of keyboard key states into the calling thread's keyboard input-state table. This is the same table accessed by the GetKeyboardState and GetKeyState functions. Changes made to this table do not affect keyboard input to any other thread.
SetLastErrorEx	Sets the last-error code.
SetLayeredWindowAttributes	Sets the opacity and transparency color key of a layered window.
SetMenu	Assigns a new menu to the specified window.
SetMenuContextHelpId	Associates a Help context identifier with a menu.
SetMenuDefaultItem	Sets the default menu item for the specified menu.
SetMenuInfo	Sets information for a specified menu.
SetMenuItemBitmaps	Associates the specified bitmap with a menu item. Whether the menu item is selected or clear, the system displays the appropriate bitmap next to the menu item.

TITLE	DESCRIPTION
SetMenuItemInfoA	Changes information about a menu item.
SetMenuItemInfoW	Changes information about a menu item.
SetMessageExtraInfo	Sets the extra message information for the current thread.
SetParent	Changes the parent window of the specified child window.
SetPhysicalCursorPos	Sets the position of the cursor in physical coordinates.
SetProcessDefaultLayout	Changes the default layout when windows are created with no parent or owner only for the currently running process.
SetProcessDPIAware	SetProcessDPIAware may be altered or unavailable. Instead, use SetProcessDPIAwareness.
SetProcessDpiAwarenessContext	Sets the current process to a specified dots per inch (dpi) awareness context. The DPI awareness contexts are from the DPI_AWARENESS_CONTEXT value.
SetProcessRestrictionExemption	Exempts the calling process from restrictions preventing desktop processes from interacting with the Windows Store app environment. This function is used by development and debugging tools.
SetProcessWindowStation	Assigns the specified window station to the calling process.
SetPropA	Adds a new entry or changes an existing entry in the property list of the specified window.
SetPropW	Adds a new entry or changes an existing entry in the property list of the specified window.
SetRect	The SetRect function sets the coordinates of the specified rectangle. This is equivalent to assigning the left, top, right, and bottom arguments to the appropriate members of the RECT structure.
SetRectEmpty	The SetRectEmpty function creates an empty rectangle in which all coordinates are set to zero.
SetScrollInfo	The SetScrollInfo function sets the parameters of a scroll bar, including the minimum and maximum scrolling positions, the page size, and the position of the scroll box (thumb). The function also redraws the scroll bar, if requested.
SetScrollPos	The SetScrollPos function sets the position of the scroll box (thumb) in the specified scroll bar and, if requested, redraws the scroll bar to reflect the new position of the scroll box.
SetScrollRange	The SetScrollRange function sets the minimum and maximum scroll box positions for the specified scroll bar.
SetSysColors	Sets the colors for the specified display elements.

TITLE	DESCRIPTION
SetSystemCursor	Enables an application to customize the system cursors. It replaces the contents of the system cursor specified by the id parameter with the contents of the cursor specified by the hcur parameter and then destroys hcur.
SetThreadDesktop	Assigns the specified desktop to the calling thread. All subsequent operations on the desktop use the access rights granted to the desktop.
SetThreadDpiAwarenessContext	Set the DPI awareness for the current thread to the provided value.
SetThreadDpiHostingBehavior	Sets the thread's DPI_HOSTING_BEHAVIOR. This behavior allows windows created in the thread to host child windows with a different DPI_AWARENESS_CONTEXT.
SetTimer	Creates a timer with the specified time-out value.
SetUserObjectInformationA	Sets information about the specified window station or desktop object.
SetUserObjectInformationW	Sets information about the specified window station or desktop object.
SetUserObjectSecurity	Sets the security of a user object. This can be, for example, a window or a DDE conversation.
SetWindowContextHelpId	Associates a Help context identifier with the specified window.
SetWindowDisplayAffinity	Stores the display affinity setting in kernel mode on the hWnd associated with the window.
SetWindowFeedbackSetting	Sets the feedback configuration for a window.
SetWindowLongA	Changes an attribute of the specified window. The function also sets the 32-bit (long) value at the specified offset into the extra window memory.
SetWindowLongPtrA	Changes an attribute of the specified window.
SetWindowLongPtrW	Changes an attribute of the specified window.
SetWindowLongW	Changes an attribute of the specified window. The function also sets the 32-bit (long) value at the specified offset into the extra window memory.
SetWindowPlacement	Sets the show state and the restored, minimized, and maximized positions of the specified window.

TITLE	DESCRIPTION
SetWindowPos	Changes the size, position, and Z order of a child, pop-up, or top-level window. These windows are ordered according to their appearance on the screen. The topmost window receives the highest rank and is the first window in the Z order.
SetWindowRgn	The SetWindowRgn function sets the window region of a window.
SetWindowsHookExA	Installs an application-defined hook procedure into a hook chain.
SetWindowsHookExW	Installs an application-defined hook procedure into a hook chain.
SetWindowTextA	Changes the text of the specified window's title bar (if it has one). If the specified window is a control, the text of the control is changed. However, SetWindowText cannot change the text of a control in another application.
SetWindowTextW	Changes the text of the specified window's title bar (if it has one). If the specified window is a control, the text of the control is changed. However, SetWindowText cannot change the text of a control in another application.
SetWinEventHook	Sets an event hook function for a range of events.
ShowCaret	Makes the caret visible on the screen at the caret's current position. When the caret becomes visible, it begins flashing automatically.
ShowCursor	Displays or hides the cursor.
ShowOwnedPopups	Shows or hides all pop-up windows owned by the specified window.
ShowScrollBar	The ShowScrollBar function shows or hides the specified scroll bar.
ShowWindow	Sets the specified window's show state.
ShowWindowAsync	Sets the show state of a window without waiting for the operation to complete.
ShutdownBlockReasonCreate	Indicates that the system cannot be shut down and sets a reason string to be displayed to the user if system shutdown is initiated.
ShutdownBlockReasonDestroy	Indicates that the system can be shut down and frees the reason string.
ShutdownBlockReasonQuery	Retrieves the reason string set by the ShutdownBlockReasonCreate function.

TITLE	DESCRIPTION
SkipPointerFrameMessages	Determines which pointer input frame generated the most recently retrieved message for the specified pointer and discards any queued (unretrieved) pointer input messages generated from the same pointer input frame.
SoundSentry	Triggers a visual signal to indicate that a sound is playing.
SubtractRect	The SubtractRect function determines the coordinates of a rectangle formed by subtracting one rectangle from another.
SwapMouseButton	Reverses or restores the meaning of the left and right mouse buttons.
SwitchDesktop	Makes the specified desktop visible and activates it. This enables the desktop to receive input from the user.
SwitchToThisWindow	Switches focus to the specified window and brings it to the foreground.
SystemParametersInfoA	Retrieves or sets the value of one of the system-wide parameters.
SystemParametersInfoForDpi	Retrieves the value of one of the system-wide parameters, taking into account the provided DPI value.
SystemParametersInfoW	Retrieves or sets the value of one of the system-wide parameters.
TabbedTextOutA	The TabbedTextOut function writes a character string at a specified location, expanding tabs to the values specified in an array of tab-stop positions. Text is written in the currently selected font, background color, and text color.
TabbedTextOutW	The TabbedTextOut function writes a character string at a specified location, expanding tabs to the values specified in an array of tab-stop positions. Text is written in the currently selected font, background color, and text color.
TileWindows	Tiles the specified child windows of the specified parent window.
ToAscii	Translates the specified virtual-key code and keyboard state to the corresponding character or characters.
ToAsciiEx	Translates the specified virtual-key code and keyboard state to the corresponding character or characters. The function translates the code using the input language and physical keyboard layout identified by the input locale identifier.
TOUCH_COORD_TO_PIXEL	Converts touch coordinates to pixels.
ToUnicode	Translates the specified virtual-key code and keyboard state to the corresponding Unicode character or characters.

TITLE	DESCRIPTION
ToUnicodeEx	Translates the specified virtual-key code and keyboard state to the corresponding Unicode character or characters.
TrackMouseEvent	Posts messages when the mouse pointer leaves a window or hovers over a window for a specified amount of time.
TrackPopupMenu	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.
TrackPopupMenuEx	Displays a shortcut menu at the specified location and tracks the selection of items on the shortcut menu. The shortcut menu can appear anywhere on the screen.
TranslateAcceleratorA	Processes accelerator keys for menu commands.
TranslateAcceleratorW	Processes accelerator keys for menu commands.
TranslateMDISysAccel	Processes accelerator keystrokes for window menu commands of the multiple-document interface (MDI) child windows associated with the specified MDI client window.
TranslateMessage	Translates virtual-key messages into character messages. The character messages are posted to the calling thread's message queue, to be read the next time the thread calls the GetMessage or PeekMessage function.
UnhookWindowsHookEx	Removes a hook procedure installed in a hook chain by the SetWindowsHookEx function.
UnhookWinEvent	Removes an event hook function created by a previous call to SetWinEventHook.
UnionRect	The UnionRect function creates the union of two rectangles. The union is the smallest rectangle that contains both source rectangles.
UnloadKeyboardLayout	Unloads an input locale identifier (formerly called a keyboard layout).
UnregisterClassA	Unregisters a window class, freeing the memory required for the class.
UnregisterClassW	Unregisters a window class, freeing the memory required for the class.
UnregisterDeviceNotification	Closes the specified device notification handle.
UnregisterHotKey	Frees a hot key previously registered by the calling thread.
UnregisterPointerInputTarget	Allows the caller to unregister a target window to which all pointer input of the specified type is redirected.

TITLE	DESCRIPTION
UnregisterPointerInputTargetEx	UnregisterPointerInputTargetEx may be altered or unavailable. Instead, use UnregisterPointerInputTarget.
UnregisterPowerSettingNotification	Unregisters the power setting notification.
UnregisterSuspendResumeNotification	Cancels a registration to receive notification when the system is suspended or resumed. Similar to PowerUnregisterSuspendResumeNotification but operates in user mode.
UnregisterTouchWindow	Registers a window as no longer being touch-capable.
UpdateLayeredWindow	Updates the position, size, shape, content, and translucency of a layered window.
UpdateWindow	The UpdateWindow function updates the client area of the specified window by sending a WM_PAINT message to the window if the window's update region is not empty.
UserHandleGrantAccess	Grants or denies access to a handle to a User object to a job that has a user-interface restriction.
ValidateRect	The ValidateRect function validates the client area within a rectangle by removing the rectangle from the update region of the specified window.
ValidateRgn	The ValidateRgn function validates the client area within a region by removing the region from the current update region of the specified window.
VkKeyScanA	Translates a character to the corresponding virtual-key code and shift state for the current keyboard.
VkKeyScanExA	Translates a character to the corresponding virtual-key code and shift state. The function translates the character using the input language and physical keyboard layout identified by the input locale identifier.
VkKeyScanExW	Translates a character to the corresponding virtual-key code and shift state. The function translates the character using the input language and physical keyboard layout identified by the input locale identifier.
VkKeyScanW	Translates a character to the corresponding virtual-key code and shift state for the current keyboard.
WaitForInputIdle	Waits until the specified process has finished processing its initial input and is waiting for user input with no input pending, or until the time-out interval has elapsed.
WaitMessage	Yields control to other threads when a thread has no other messages in its message queue. The WaitMessage function suspends the thread and does not return until a new message is placed in the thread's message queue.

TITLE	DESCRIPTION
WindowFromDC	The WindowFromDC function returns a handle to the window associated with the specified display device context (DC). Output functions that use the specified device context draw into this window.
WindowFromPhysicalPoint	Retrieves a handle to the window that contains the specified physical point.
WindowFromPoint	Retrieves a handle to the window that contains the specified point.
WinHelpA	Launches Windows Help (Winhelp.exe) and passes additional data that indicates the nature of the help requested by the application.
WinHelpW	Launches Windows Help (Winhelp.exe) and passes additional data that indicates the nature of the help requested by the application.
wsprintfA	Writes formatted data to the specified buffer.
wsprintfW	Writes formatted data to the specified buffer.
wvsprintfA	Writes formatted data to the specified buffer using a pointer to a list of arguments.
wvsprintfW	Writes formatted data to the specified buffer using a pointer to a list of arguments.

Callback functions

TITLE	DESCRIPTION
DLGPROC	Application-defined callback function used with the CreateDialog and DialogBox families of functions.
DRAWSTATEPROC	The DrawStateProc function is an application-defined callback function that renders a complex image for the DrawState function.
EDITWORDBREAKPROCA	An application-defined callback function used with the EM_SETWORDBREAKPROC message.
EDITWORDBREAKPROCW	An application-defined callback function used with the EM_SETWORDBREAKPROC message.
GRAYSTRINGPROC	The OutputProc function is an application-defined callback function used with the GrayString function.

TITLE	DESCRIPTION
HOOKPROC	An application-defined or library-defined callback function used with the SetWindowsHookEx function. The system calls this function after the SendMessage function is called. The hook procedure can examine the message; it cannot modify it.
MONITORENUMPROC	A MonitorEnumProc function is an application-defined callback function that is called by the EnumDisplayMonitors function.
PROPENUMPROCA	An application-defined callback function used with the EnumProps function.
PROPENUMPROCEXA	Application-defined callback function used with the EnumPropsEx function.
PROPENUMPROCEXW	Application-defined callback function used with the EnumPropsEx function.
PROPENUMPROCW	An application-defined callback function used with the EnumProps function.
SENDASYNCPROC	An application-defined callback function used with the SendMessageCallback function.
TIMERPROC	An application-defined callback function that processes WM_TIMER messages. The TIMERPROC type defines a pointer to this callback function. TimerProc is a placeholder for the application-defined function name.
WINEVENTPROC	An application-defined callback (or hook) function that the system calls in response to events generated by an accessible object.

Structures

TITLE	DESCRIPTION
ACCEL	Defines an accelerator key used in an accelerator table.
ACCESSTIMEOUT	Contains information about the time-out period associated with the accessibility features.
ALTTABINFO	Contains status information for the application-switching (ALT+TAB) window.
ANIMATIONINFO	Describes the animation effects associated with user actions.
AUDIODESCRIPTION	Contains information associated with audio descriptions. This structure is used with the SystemParametersInfo function when the SPI_GETAUDIODESCRIPTION or SPI_SETAUDIODESCRIPTION action value is specified.

TITLE	DESCRIPTION
BSMINFO	Contains information about a window that denied a request from BroadcastSystemMessageEx.
CBT_CREATEWENDA	Contains information passed to a WH_CBT hook procedure, CBTProc, before a window is created.
CBT_CREATEWNDW	Contains information passed to a WH_CBT hook procedure, CBTProc, before a window is created.
CBTACTIVATESTRUCT	Contains information passed to a WH_CBT hook procedure, CBTProc, before a window is activated.
CHANGEFILTERSTRUCT	Contains extended result information obtained by calling the ChangeWindowMessageFilterEx function.
CLIENTCREATESTRUCT	Contains information about the menu and first multiple-document interface (MDI) child window of an MDI client window.
COMBOBOXINFO	Contains combo box status information.
COMPAREITEMSTRUCT	Supplies the identifiers and application-supplied data for two items in a sorted, owner-drawn list box or combo box.
COPYDATASTRUCT	Contains data to be passed to another application by the WM_COPYDATA message.
CREATESTRUCTA	Defines the initialization parameters passed to the window procedure of an application. These members are identical to the parameters of the CreateWindowEx function.
CREATESTRUCTW	Defines the initialization parameters passed to the window procedure of an application. These members are identical to the parameters of the CreateWindowEx function.
CURSORSINFO	Contains global cursor information.
CURSORSHAPE	Contains information about a cursor.
CWPRETSTRUCT	Defines the message parameters passed to a WH_CALLWNDPROCRET hook procedure, CallWndRetProc.
CWPSTRUCT	Defines the message parameters passed to a WH_CALLWNDPROC hook procedure, CallWndProc.
DEBUGHOOKINFO	Contains debugging information passed to a WH_DEBUG hook procedure, DebugProc.
DELETEITEMSTRUCT	Describes a deleted list box or combo box item.
DLGITEMTEMPLATE	Defines the dimensions and style of a control in a dialog box. One or more of these structures are combined with a DLGTEMPLATE structure to form a standard template for a dialog box.

TITLE	DESCRIPTION
DLGTEMPLATE	Defines the dimensions and style of a dialog box.
DRAWITEMSTRUCT	Provides information that the owner window uses to determine how to paint an owner-drawn control or menu item.
DRAWTEXTPARAMS	The DRAWTEXTPARAMS structure contains extended formatting options for the DrawTextEx function.
EVENTMSG	Contains information about a hardware message sent to the system message queue. This structure is used to store message information for the JournalPlaybackProc callback function.
FILTERKEYS	Contains information about the FilterKeys accessibility feature, which enables a user with disabilities to set the keyboard repeat rate (RepeatKeys), acceptance delay (SlowKeys), and bounce rate (BounceKeys).
FLASHWINFO	Contains the flash status for a window and the number of times the system should flash the window.
GESTURECONFIG	Gets and sets the configuration for enabling gesture messages and the type of this configuration.
GESTUREINFO	Stores information about a gesture.
GESTURENOTIFYSTRUCT	When transmitted with WM_GESTURENOTIFY messages, passes information about a gesture.
GUITHREADINFO	Contains information about a GUI thread.
HARDWAREINPUT	Contains information about a simulated message generated by an input device other than a keyboard or mouse.
HELPINFO	Contains information about an item for which context-sensitive help has been requested.
HELPWININFOA	Contains the size and position of either a primary or secondary Help window. An application can set this information by calling the WinHelp function with the HELP_SETWINPOS value.
HELPWININFOW	Contains the size and position of either a primary or secondary Help window. An application can set this information by calling the WinHelp function with the HELP_SETWINPOS value.
HIGHCONTRASTA	Contains information about the high contrast accessibility feature.
HIGHCONTRASTW	Contains information about the high contrast accessibility feature.

TITLE	DESCRIPTION
ICONINFO	Contains information about an icon or a cursor.
ICONINFOEXA	Contains information about an icon or a cursor. Extends ICONINFO. Used by GetIconInfoEx.
ICONINFOEXW	Contains information about an icon or a cursor. Extends ICONINFO. Used by GetIconInfoEx.
ICONMETRICS_A	Contains the scalable metrics associated with icons. This structure is used with the SystemParametersInfo function when the SPI_GETICONMETRICS or SPI_SETICONMETRICS action is specified.
ICONMETRICS_W	Contains the scalable metrics associated with icons. This structure is used with the SystemParametersInfo function when the SPI_GETICONMETRICS or SPI_SETICONMETRICS action is specified.
INPUT	Used by SendInput to store information for synthesizing input events such as keystrokes, mouse movement, and mouse clicks.
INPUT_INJECTION_VALUE	Contains the input injection details.
INPUT_MESSAGE_SOURCE	Contains information about the source of the input message.
INPUT_TRANSFORM	Defines the matrix that represents a transform on a message consumer.
KBDLLHOOKSTRUCT	Contains information about a low-level keyboard input event.
KEYBDINPUT	Contains information about a simulated keyboard event.
LASTINPUTINFO	Contains the time of the last input.
MDICREATESTRUCT_A	Contains information about the class, title, owner, location, and size of a multiple-document interface (MDI) child window.
MDICREATESTRUCT_W	Contains information about the class, title, owner, location, and size of a multiple-document interface (MDI) child window.
MDINEXTMENU	Contains information about the menu to be activated.
MEASUREITEMSTRUCT	Informs the system of the dimensions of an owner-drawn control or menu item. This allows the system to process user interaction with the control correctly.
MENUBARINFO	Contains menu bar information.

TITLE	DESCRIPTION
MENUGETOBJECTINFO	Contains information about the menu that the mouse cursor is on.
MENUINFO	Contains information about a menu.
MENUITEMINFOA	Contains information about a menu item.
MENUITEMINFOW	Contains information about a menu item.
MENUITEMTEMPLATE	Defines a menu item in a menu template.
MENUITEMTEMPLATEHEADER	Defines the header for a menu template. A complete menu template consists of a header and one or more menu item lists.
MINIMIZEDMETRICS	Contains the scalable metrics associated with minimized windows.
MINMAXINFO	Contains information about a window's maximized size and position and its minimum and maximum tracking size.
MONITORINFO	The MONITORINFO structure contains information about a display monitor. The GetMonitorInfo function stores information in a MONITORINFO structure or a MONITORINFOEX structure. The MONITORINFO structure is a subset of the MONITORINFOEX structure.
MONITORINFOEXA	The MONITORINFOEX structure contains information about a display monitor. The GetMonitorInfo function stores information into a MONITORINFOEX structure or a MONITORINFO structure. The MONITORINFOEX structure is a superset of the MONITORINFO structure.
MONITORINFOEXW	The MONITORINFOEX structure contains information about a display monitor. The GetMonitorInfo function stores information into a MONITORINFOEX structure or a MONITORINFO structure. The MONITORINFOEX structure is a superset of the MONITORINFO structure.
MOUSEHOOKSTRUCT	Contains information about a mouse event passed to a WH_MOUSE hook procedure, MouseProc.
MOUSEHOOKSTRUCTEX	Contains information about a mouse event passed to a WH_MOUSE hook procedure, MouseProc. This is an extension of the MOUSEHOOKSTRUCT structure that includes information about wheel movement or the use of the X button.
MOUSEINPUT	Contains information about a simulated mouse event.
MOUSEKEYS	Contains information about the MouseKeys accessibility feature.
MOUSEMOVEPOINT	Contains information about the mouse's location in screen coordinates.

TITLE	DESCRIPTION
MSG	Contains message information from a thread's message queue.
MSGBOXPARAMSA	Contains information used to display a message box. The MessageBoxIndirect function uses this structure.
MSGBOXPARAMSW	Contains information used to display a message box. The MessageBoxIndirect function uses this structure.
MSLLHOOKSTRUCT	Contains information about a low-level mouse input event.
MULTIKEYHELPA	Specifies a keyword to search for and the keyword table to be searched by Windows Help.
MULTIKEYHELPW	Specifies a keyword to search for and the keyword table to be searched by Windows Help.
NCCALCSIZE_PARAMS	Contains information that an application can use while processing the WM_NCCALCSIZE message to calculate the size, position, and valid contents of the client area of a window.
NMHDR	Contains information about a notification message.
NONCLIENTMETRICS	Contains the scalable metrics associated with the nonclient area of a nonminimized window.
NONCLIENTMETRICSW	Contains the scalable metrics associated with the nonclient area of a nonminimized window.
PAINTSTRUCT	The PAINTSTRUCT structure contains information for an application. This information can be used to paint the client area of a window owned by that application.
POINTER_DEVICE_CURSOR_INFO	Contains cursor ID mappings for pointer devices.
POINTER_DEVICE_INFO	Contains information about a pointer device. An array of these structures is returned from the GetPointerDevices function. A single structure is returned from a call to the GetPointerDevice function.
POINTER_DEVICE_PROPERTY	Contains pointer-based device properties (Human Interface Device (HID) global items that correspond to HID usages).
POINTER_INFO	Contains basic pointer information common to all pointer types. Applications can retrieve this information using the GetPointerInfo, GetPointerFrameInfo, GetPointerInfoHistory and GetPointerFrameInfoHistory functions.
POINTER_PEN_INFO	Defines basic pen information common to all pointer types.

TITLE	DESCRIPTION
POINTER_TOUCH_INFO	Defines basic touch information common to all pointer types.
POINTER_TYPE_INFO	Contains information about the pointer input type.
POWERBROADCAST_SETTING	Sent with a power setting event and contains data about the specific change.
RAWHID	Describes the format of the raw input from a Human Interface Device (HID).
RAWINPUT	Contains the raw input from a device.
RAWINPUTDEVICE	Defines information for the raw input devices.
RAWINPUTDEVICELIST	Contains information about a raw input device.
RAWINPUTHEADER	Contains the header information that is part of the raw input data.
RAWKEYBOARD	Contains information about the state of the keyboard.
RAWMOUSE	Contains information about the state of the mouse.
RID_DEVICE_INFO	Defines the raw input data coming from any device.
RID_DEVICE_INFO_HID	Defines the raw input data coming from the specified Human Interface Device (HID).
RID_DEVICE_INFO_KEYBOARD	Defines the raw input data coming from the specified keyboard.
RID_DEVICE_INFO_MOUSE	Defines the raw input data coming from the specified mouse.
SCROLLBARINFO	The SCROLLBARINFO structure contains scroll bar information.
SCROLLINFO	The SCROLLINFO structure contains scroll bar parameters to be set by the SetScrollInfo function (or SBM_SETSCROLLINFO message), or retrieved by the GetScrollInfo function (or SBM_GETSCROLLINFO message).
SERIALKEYSA	Contains information about the SerialKeys accessibility feature, which interprets data from a communication aid attached to a serial port as commands causing the system to simulate keyboard and mouse input.
SERIALKEYSW	Contains information about the SerialKeys accessibility feature, which interprets data from a communication aid attached to a serial port as commands causing the system to simulate keyboard and mouse input.

TITLE	DESCRIPTION
SOUNDSENTRYA	Contains information about the SoundSentry accessibility feature. When the SoundSentry feature is on, the computer displays a visual indication only when a sound is generated.
SOUNDSENTRYW	Contains information about the SoundSentry accessibility feature. When the SoundSentry feature is on, the computer displays a visual indication only when a sound is generated.
STICKYKEYS	Contains information about the StickyKeys accessibility feature.
STYLESTRUCT	Contains the styles for a window.
TITLEBARINFO	Contains title bar information.
TITLEBARINFOEX	Expands on the information described in the TITLEBARINFO structure by including the coordinates of each element of the title bar.
TOGGLEKEYS	Contains information about the ToggleKeys accessibility feature.
TOUCH_HIT_TESTING_INPUT	Contains information about the touch contact area reported by the touch digitizer.
TOUCH_HIT_TESTING_PROXIMITY_EVALUATION	Contains the hit test score that indicates whether the object is the likely target of the touch contact area, relative to other objects that intersect the touch contact area.
TOUCHINPUT	Encapsulates data for touch input.
TOUCHPREDICTIONPARAMETERS	Contains hardware input details that can be used to predict touch targets and help compensate for hardware latency when processing touch and gesture input that contains distance and velocity data.
TPMPARAMS	Contains extended parameters for the TrackPopupMenuEx function.
TRACKMOUSEEVENT	Used by the TrackMouseEvent function to track when the mouse pointer leaves a window or hovers over a window for a specified amount of time.
UPDATELAYEREDWINDOWINFO	Used by UpdateLayeredWindowIndirect to provide position, size, shape, content, and translucency information for a layered window.
USAGE_PROPERTIES	Contains device properties (Human Interface Device (HID) global items that correspond to HID usages) for any type of HID input device.
USEROBJECTFLAGS	Contains information about a window station or desktop handle.

TITLE	DESCRIPTION
WINDOWINFO	Contains window information.
WINDOWPLACEMENT	Contains information about the placement of a window on the screen.
WINDOWPOS	Contains information about the size and position of a window.
WNDCLASSA	Contains the window class attributes that are registered by the RegisterClass function.
WNDCLASSEXA	Contains window class information.
WNDCLASSEXW	Contains window class information.
WNDCLASSW	Contains the window class attributes that are registered by the RegisterClass function.
WTSSession_Notification	Provides information about the session change notification. A service receives this structure in its HandlerEx function in response to a session change event.

Enumerations

TITLE	DESCRIPTION
AR_STATE	Indicates the state of screen auto-rotation for the system. For example, whether auto-rotation is supported, and whether it is enabled by the user.
DIALOG_CONTROL_DPI_CHANGE_BEHAVIORS	Describes per-monitor DPI scaling behavior overrides for child windows within dialogs. The values in this enumeration are bitfields and can be combined.
DIALOG_DPI_CHANGE_BEHAVIORS	In Per Monitor v2 contexts, dialogs will automatically respond to DPI changes by resizing themselves and re-computing the positions of their child windows (here referred to as re-layouting).
FEEDBACK_TYPE	Specifies the visual feedback associated with an event.
INPUT_MESSAGE_DEVICE_TYPE	The type of device that sent the input message.
INPUT_MESSAGE_ORIGIN_ID	The ID of the input message source.
ORIENTATION_PREFERENCE	Indicates the screen orientation preference for a desktop app process.
POINTER_BUTTON_CHANGE_TYPE	Identifies a change in the state of a button associated with a pointer.
POINTER_DEVICE_CURSOR_TYPE	Identifies the pointer device cursor types.

TITLE	DESCRIPTION
POINTER_DEVICE_TYPE	Identifies the pointer device types.
POINTER_FEEDBACK_MODE	Identifies the visual feedback behaviors available to CreateSyntheticPointerDevice.
tagPOINTER_INPUT_TYPE	Identifies the pointer input types.