This is part of **Win16 API** which allow to create versions of program from one source code to run under OS/2 and Win16. Under OS/2 program can be running under Win-OS/2 if program is Windows NE executable, and with help on Windows Libraries for OS/2, if it is OS/2 NE executable. Here is a WLO to OS/2 API mapping draft

2021/09/01 04:23 · prokushev · 0 Comments

Ordinal	Name	Description	Status	Version
1	MESSAGEBOX	Displays a modal dialog box that contains a system icon, a set of buttons, and a brief application-specific message		
2	OLDEXITWINDOWS		Done	
5	INITAPP	Initializes an application instance		
6	POSTQUITMESSAGE	Indicates that the application has requested to terminate		
7	EXITWINDOWS	Shuts down Windows and restarts the operating system		
10	SETTIMER	Creates a timer with the specified time-out value		
11	BEAR11			
12	KILLTIMER	Destroys the specified timer		
13	GETTICKCOUNT	Retrieves the number of milliseconds that have elapsed since the system was started	Done	
14	GETTIMERRESOLUTION	Retrieves the minimum timer resolution for the system		
15	GETCURRENTTIME	Retrieves the current Windows time	Done	
16	CLIPCURSOR	Confines the cursor to a rectangular area on the screen		
17	GETCURSORPOS	Retrieves the position of the mouse cursor, in screen coordinates		
18	SETCAPTURE	Sets the mouse capture to the specified window		
19	RELEASECAPTURE	Releases the mouse capture from a window and restores normal mouse input processing		
20	SETDOUBLECLICKTIME	Sets the double-click time for the mouse		
21	GETDOUBLECLICKTIME	Retrieves the current double-click time for the mouse		
22	SETFOCUS	Sets the keyboard focus to the specified window		
23	GETFOCUS	Retrieves the handle to the window that has the keyboard focus		
24	REMOVEPROP	Removes a property from a window's property list		
25	GETPROP	Retrieves a data handle from a window's property list		

Ordinal	Name	Description	Status	Version
26	SETPROP	Adds a new entry or changes an existing entry in a window's property list		
27	ENUMPROPS	Enumerates all entries in a window's property list		
28	CLIENTTOSCREEN	Converts the client-area coordinates of a specified point to screen coordinates		
29	SCREENTOCLIENT	Converts the screen coordinates of a specified point to client coordinates		
30	WINDOWFROMPOINT	Retrieves the handle of the window that contains the specified point		
31	ISICONIC	Determines whether the specified window is minimized (iconic)		
32	GETWINDOWRECT	Retrieves the dimensions of the bounding rectangle of the specified window		
33	GETCLIENTRECT	Retrieves the coordinates of a window's client area		
34	ENABLEWINDOW	Enables or disables mouse and keyboard input to the specified window		
35	ISWINDOWENABLED	Determines whether the specified window is enabled for mouse and keyboard input		
36	GETWINDOWTEXT	Copies the text of the specified window's title bar into a buffer	Done	
37	SETWINDOWTEXT	Changes the text of the specified window's title bar	Done	
38	GETWINDOWTEXTLENGTH	Retrieves the length of the specified window's title bar text	Done	
39	BEGINPAINT	Prepares the specified window for painting and fills a PAINTSTRUCT structure with painting information		
40	ENDPAINT	Marks the end of painting in the specified window		
41	CREATEWINDOW	Creates an overlapped, pop-up, or child window	Done	
42	SHOWWINDOW	Sets the specified window's show state		
43	CLOSEWINDOW	Minimizes (but does not destroy) the specified window		
44	OPENICON	Restores a minimized (iconic) window to its previous size and position		
45	BRINGWINDOWTOTOP	Brings the specified window to the top of the Z order		
46	GETPARENT	Retrieves the handle of the specified window's parent window		
47	ISWINDOW	Determines whether the specified handle is a window handle		
48	ISCHILD	Determines whether a window is a child window of the specified parent window		

Ordinal	Name	Description	Status	Version
49	ISWINDOWVISIBLE	Determines whether the specified window is visible		
50	FINDWINDOW	Retrieves the handle to the top-level window whose class name and window name match the specified strings		
52	ANYPOPUP	Indicates whether an owned, visible, top-level pop-up, or overlapped window exists on the screen		
53	DESTROYWINDOW	Destroys the specified window		
54	ENUMWINDOWS	Enumerates all top-level windows on the screen		
55	ENUMCHILDWINDOWS	Enumerates the child windows that belong to the specified parent window		
56	MOVEWINDOW	Changes the position and dimensions of the specified window		
57	REGISTERCLASS	Registers a window class for subsequent use in calls to the CreateWindow function	Done	
58	GETCLASSNAME	Retrieves the name of the class to which the specified window belongs		
59	SETACTIVEWINDOW	Sets the specified window to the active window		
60	GETACTIVEWINDOW	Retrieves the handle to the active window		
61	SCROLLWINDOW	Scrolls the contents of the specified window's client area		
62	SETSCROLLPOS	Sets the position of the scroll box (thumb) in the specified scroll bar		
63	GETSCROLLPOS	Retrieves the current position of the scroll box (thumb) in the specified scroll bar		
64	SETSCROLLRANGE	Sets the minimum and maximum scroll box positions for the specified scroll bar		
65	GETSCROLLRANGE	Retrieves the current minimum and maximum scroll box positions for the specified scroll bar		
66	GETDC	Retrieves a handle to a device context for the client area of the specified window		
67	GETWINDOWDC	Retrieves a handle to a device context for the entire window, including title bar, menus, and scroll bars		
68	RELEASEDC	Releases a device context, freeing it for use by other applications		
69	SETCURSOR	Sets the cursor shape		
70	SETCURSORPOS	Moves the cursor to the specified screen coordinates		
71	SHOWCURSOR	Displays or hides the cursor		
72	SETRECT	Sets the coordinates of the specified rectangle	Done	

Ordinal	Name	Description	Status	Version
73	SETRECTEMPTY	Creates an empty rectangle	Done	
74	COPYRECT	Copies the coordinates of one rectangle to another	Done	
75	ISRECTEMPTY	Determines whether the specified rectangle is empty	Done	
76	PTINRECT	Determines whether the specified point is inside the specified rectangle	Done	
77	OFFSETRECT	Moves the specified rectangle by the specified offsets	Done	
78	INFLATERECT	Increases or decreases the width and height of the specified rectangle	Done	
79	INTERSECTRECT	Calculates the intersection of two source rectangles and places the coordinates of the intersection rectangle into the destination rectangle	Done	
80	UNIONRECT	Creates the union of two rectangles	Done	
81	FILLRECT	Fills a rectangle by using the specified brush	Done	
82	INVERTRECT	Inverts a rectangle in a window by performing a logical NOT operation on the color values for each pixel in the rectangle's interior	Done	
83	FRAMERECT	Draws a border around the specified rectangle by using the specified brush	Done	
84	DRAWICON	Draws an icon or cursor into the specified device context	Done	
85	DRAWTEXT	Draws formatted text in the specified rectangle	Done	
86	BEAR86			
87	DIALOGBOX	Creates a modal dialog box from a dialog box template resource	Done	
88	ENDDIALOG	Destroys a modal dialog box, causing the system to end any processing for the dialog box		
89	CREATEDIALOG	Creates a modeless dialog box from a dialog box template resource	Done	
90	ISDIALOGMESSAGE	Determines whether a message is intended for the specified dialog box and, if it is, processes the message		
91	GETDLGITEM	Retrieves the handle of a control in the specified dialog box		
92	SETDLGITEMTEXT	Sets the title or text of a control in a dialog box	Done	
93	GETDLGITEMTEXT	Retrieves the title or text associated with a control in a dialog box	Done	
94	SETDLGITEMINT	Sets the text of a control in a dialog box to the string representation of a specified integer value		

Ordinal	Name	Description	Status	Version
95	GETDLGITEMINT	Retrieves the text of the specified control by converting it to an integer value		
96	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		
97	CHECKDLGBUTTON	Changes the check state of a button control	Done	
98	ISDLGBUTTONCHECKED	Determines whether a button control is checked or not	Done	
99	DLGDIRSELECT	Retrieves the current selection from a list box filled by the DlgDirList function	Done	
100	DLGDIRLIST	Fills a list box with a file or directory listing		
101	SENDDLGITEMMESSAGE	Sends a message to the specified control in a dialog box	Done	
102	ADJUSTWINDOWRECT	Calculates the required size of the window rectangle based on the desired client rectangle size		
103	MAPDIALOGRECT	Converts the dialog-box coordinates of a specified rectangle to screen coordinates		
104	MESSAGEBEEP	Plays a waveform sound corresponding to the specified alert type		
105	FLASHWINDOW	Flashes the specified window one time		
106	GETKEYSTATE	Retrieves the status of the specified virtual key		
107	DEFWINDOWPROC	Calls the default window procedure to provide default processing for any window messages that an application does not process		
108	GETMESSAGE	Retrieves a message from the calling thread's message queue		
109	PEEKMESSAGE	Checks the thread message queue for a message and returns immediately		
110	POSTMESSAGE	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		
111	SENDMESSAGE	Sends the specified message to a window or windows and does not return until the window procedure has processed the message		
112	WAITMESSAGE	Yields control to other threads when a thread has no other messages in its message queue		
113	TRANSLATEMESSAGE	Translates virtual-key messages into character messages		

Ordinal	Name	Description	Status	Version
114	DISPATCHMESSAGE	Dispatches a message to a window procedure		
115	REPLYMESSAGE	Used to reply to a message sent through the SendMessage function without returning control to the function that called SendMessage		
116	POSTAPPMESSAGE	Places a message in the message queue of all top-level windows and returns immediately		
118	REGISTERWINDOWMESSAGE	Defines a new window message that is guaranteed to be unique throughout the system		
119	GETMESSAGEPOS	Retrieves the cursor position for the last message retrieved by the GetMessage function		
120	GETMESSAGETIME	Retrieves the message time for the last message retrieved by the GetMessage function		
121	SETWINDOWSHOOK	Installs an application-defined hook procedure into a hook chain	Done	
122	CALLWINDOWPROC	Passes message information to the specified window procedure		
123	CALLMSGFILTER	Passes a message and hook code to the hook procedures associated with the WH_SYSMSGFILTER and WH_MSGFILTER hooks		
124	UPDATEWINDOW	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	Done	
125	INVALIDATERECT	Adds a rectangle to the specified window's update region, invalidating the rectangle	Done	
126	INVALIDATERGN	Adds a region to the specified window's update region, invalidating the region	Done	
127	VALIDATERECT	Removes a rectangle from the specified window's update region, validating the rectangle	Done	
128	VALIDATERGN	Removes a region from the specified window's update region, validating the region	Done	
129	GETCLASSWORD	Retrieves the address of the specified window class's menu name string		
130	SETCLASSWORD	Replaces the address of the menu name string for the specified window class		
131	GETCLASSLONG	Retrieves the specified 32-bit value from the window class structure		
132	SETCLASSLONG	Replaces the specified 32-bit value in the window class structure		

Ordinal	Name	Description	Status	Version
133	GETWINDOWWORD	Retrieves the specified 16-bit value from the window structure		
134	SETWINDOWWORD	Replaces the specified 16-bit value in the window structure		
135	GETWINDOWLONG	Retrieves the specified 32-bit value from the window structure		
136	SETWINDOWLONG	Replaces the specified 32-bit value in the window structure		
137	OPENCLIPBOARD	Opens the clipboard for examination and prevents other applications from modifying the clipboard content		
138	CLOSECLIPBOARD	Closes the clipboard		
139	EMPTYCLIPBOARD	Empties the clipboard and frees handles to data in the clipboard		
140	GETCLIPBOARDOWNER	Retrieves the window handle of the current owner of the clipboard		
141	SETCLIPBOARDDATA	Places data on the clipboard in a specified clipboard format		
142	GETCLIPBOARDDATA	Retrieves data from the clipboard in a specified format		
143	COUNTCLIPBOARDFORMATS	Retrieves the number of different data formats currently on the clipboard		
144	ENUMCLIPBOARDFORMATS	Enumerates the data formats currently available on the clipboard		
145	REGISTERCLIPBOARDFORMAT	Registers a new clipboard format		
146	GETCLIPBOARDFORMATNAME	Retrieves the name of the specified registered clipboard format		
147	SETCLIPBOARDVIEWER	Adds the specified window to the chain of clipboard viewers		
148	GETCLIPBOARDVIEWER	Retrieves the handle to the first window in the clipboard viewer chain		
149	CHANGECLIPBOARDCHAIN	Removes a window from the chain of clipboard viewers		
150	LOADMENU	Loads the specified menu resource from the executable file associated with an application instance	Done	
151	CREATEMENU	Creates a menu	Done	
152	DESTROYMENU	Destroys the specified menu and frees any memory that the menu occupies	Done	
153	CHANGEMENU	Adds, deletes, or modifies menu items	Done	
154	CHECKMENUITEM	Adds or removes a check mark from a menu item	Done	
155	ENABLEMENUITEM	Enables, disables, or grays a menu item	Done	
156	GETSYSTEMMENU	Allows the application to access the window menu (system menu) for copying and modification	Done	
157	GETMENU	Retrieves the handle of the menu assigned to the specified window	Done	

Ordinal	Name	Description	Status	Version
158	SETMENU	Assigns a new menu to the specified window	Done	
159	GETSUBMENU	Retrieves the handle of a submenu in the specified menu	Done	
160	DRAWMENUBAR	Redraws the menu bar of the specified window	Done	
161	GETMENUSTRING	Copies the text string of the specified menu item into a buffer	Done	
162	HILITEMENUITEM	Highlights or removes the highlighting from a menu item	Done	
163	CREATECARET	Creates a new shape for the system caret and assigns ownership of the caret to the specified window	Done	
164	DESTROYCARET	Destroys the current shape of the caret and frees the caret from the window	Done	
165	SETCARETPOS	Moves the caret to the specified coordinates	Done	
166	HIDECARET	Removes the caret from the screen	Done	
167	SHOWCARET	Makes the caret visible on the screen at the caret's current position	Done	
168	SETCARETBLINKTIME	Sets the caret blink time	Done	
169	GETCARETBLINKTIME	Retrieves the caret blink time	Done	
170	ARRANGEICONICWINDOWS	Arranges all the minimized (iconic) child windows of the specified parent window		
171	WINHELP	Starts Windows Help		
173	LOADCURSOR	Loads the specified cursor resource from the executable file associated with an application instance		
174	LOADICON	Loads the specified icon resource from the executable file associated with an application instance		
175	LOADBITMAP	Loads the specified bitmap resource from the executable file associated with an application instance		
176	LOADSTRING	Loads a string resource from the executable file associated with an application instance	Done	
177	LOADACCELERATORS	Loads the specified accelerator table	Done	
178	TRANSLATEACCELERATOR	Processes accelerator keys for menu commands		
179	GETSYSTEMMETRICS	Retrieves the system metrics for the current display	Done	
180	GETSYSCOLOR	Retrieves the current color of the specified display element	Done	
181	SETSYSCOLORS	Sets the colors for one or more display elements	Done	
182	BEAR182			
183	GETCARETPOS	Retrieves the current position of the caret	Done	

Ordinal	Name	Description	Status	Version
184	QUERYSENDMESSAGE	Determines whether the message was sent from another task using the SendMessage function		
185	GRAYSTRING	Draws gray text at the specified location	Done	
186	SWAPMOUSEBUTTON	Reverses or restores the meaning of the left and right mouse buttons		
188	SETSYSMODALWINDOW	Causes the system to direct all user input to the specified window, regardless of which application the user is interacting with		
189	GETSYSMODALWINDOW	Retrieves the system-modal window, if one exists		
190	GETUPDATERECT	Retrieves the coordinates of the rectangle that completely encloses the update region of the specified window		
191	CHILDWINDOWFROMPOINT	Determines which, if any, of the child windows belonging to the specified parent window contains the specified point		
192	INSENDMESSAGE	Determines whether the current window procedure is processing a message sent from another task by the SendMessage function		
193	ISCLIPBOARDFORMATAVAILABLE	Determines whether the clipboard contains data in the specified format		
194	DLGDIRSELECTCOMBOBOX	Retrieves the current selection from a combo box filled by the DlgDirListComboBox function	Done	
195	DLGDIRLISTCOMBOBOX	Fills a combo box with a file or directory listing		
196	TABBEDTEXTOUT	Writes a character string at a specified location, expanding tabs to the values specified in an array of tab-stop positions		
197	GETTABBEDTEXTEXTENT	Computes the width and height of a character string, which may include tab characters		
198	CASCADECHILDWINDOWS	Cascades the specified child windows of the specified parent window		
199	TILECHILDWINDOWS	Tiles the specified child windows of the specified parent window		
200	OPENCOMM	Opens a communications device	Stub	
201	SETCOMMSTATE	Configures a communications device according to the specifications in a device-control block	Stub	
202	GETCOMMSTATE	Retrieves the current control settings for a communications device	Stub	
203	GETCOMMERROR	Retrieves information about the most recent communications error for a communications device	Stub	

Ordinal	Name	Description	Status	Version
204	READCOMM	Reads data from a communications device	Stub	
205	WRITECOMM	Writes data to a communications device	Stub	
206	TRANSMITCOMMCHAR	Transmits a character ahead of any pending data in the output buffer of a communications device	Stub	
207	CLOSECOMM	Closes a communications device	Stub	
208	SETCOMMEVENTMASK	Specifies a set of events to be monitored for a communications device	Stub	
209	GETCOMMEVENTMASK	Retrieves the event mask for a communications device	Stub	
210	SETCOMMBREAK	Suspends character transmission for a communications device and places the transmission line in a break state	Stub	
211	CLEARCOMMBREAK	Restores character transmission for a communications device and places the transmission line in a nonbreak state	Stub	
212	UNGETCOMMCHAR	Places a character back into the input buffer of a communications device	Stub	
213	BUILDCOMMDCB	Fills a device-control block with values specified in a string	Stub	
214	ESCAPECOMMFUNCTION	Directs a communications device to perform an extended function	Stub	
215	FLUSHCOMM	Flushes the input or output buffer of a communications device	Stub	
216	UserSeeUserDo		Done	
218	DIALOGBOXINDIRECT	Creates a modal dialog box from a dialog box template in memory	Done	
219	CREATEDIALOGINDIRECT	Creates a modeless dialog box from a dialog box template in memory	Done	
220	LOADMENUINDIRECT	Loads the specified menu template from memory	Done	
221	SCROLLDC	Scrolls a rectangle of bits horizontally and vertically		
222	GETKEYBOARDSTATE	Retrieves the status of each virtual key on the keyboard		
223	SETKEYBOARDSTATE	Sets the status of each virtual key on the keyboard		
224	GETWINDOWTASK	Retrieves the handle of the task that owns the specified window		
225	ENUMTASKWINDOWS	Enumerates all windows associated with a task		
226	LOCKINPUT	Locks or unlocks input to the specified task		
227	GETNEXTDLGGROUPITEM	Retrieves the handle of the first control in a group of controls that precedes or follows the specified control		

Ordinal	Name	Description	Status	Version
228	GETNEXTDLGTABITEM	Retrieves the handle of the first control that has the WS_TABSTOP style that precedes or follows the specified control		
229	GETTOPWINDOW	Examines the Z order of the child windows associated with the specified parent window and retrieves the handle of the child window at the top of the Z order		
230	GETNEXTWINDOW	Retrieves the handle of the next or previous window in the Z order		
231	GETSYSTEMDEBUGSTATE	Retrieves information about the current state of the system for debugging purposes		
232	SETWINDOWPOS	Changes the size, position, and Z order of a child, pop-up, or top-level window		
233	SETPARENT	Changes the parent window of the specified child window		
234	UNHOOKWINDOWSHOOK	Removes a hook procedure installed in a hook chain by the SetWindowsHook function		
235	DEFHOOKPROC	Passes a hook message to the next hook procedure in the current hook chain	Done	
236	GETCAPTURE	Retrieves the handle of the window that has captured the mouse		
237	GETUPDATERGN	Retrieves the update region of a window by copying it into the specified region		
238	EXCLUDEUPDATERGN	Prevents drawing within the update region of the specified window by excluding the update region from the specified device context's clipping region		
239	DIALOGBOXPARAM	Creates a modal dialog box from a dialog box template resource, passing application-defined data to the dialog box procedure		
240	DIALOGBOXINDIRECTPARAM	Creates a modal dialog box from a dialog box template in memory, passing application-defined data to the dialog box procedure		
241	CREATEDIALOGPARAM	Creates a modeless dialog box from a dialog box template resource, passing application-defined data to the dialog box procedure	Done	
242	CREATEDIALOGINDIRECTPARAM	Creates a modeless dialog box from a dialog box template in memory, passing application-defined data to the dialog box procedure		
243	GETDIALOGBASEUNITS	Retrieves the dialog base units used to create the dialog box		

Ordinal	Name	Description	Status	Version
244	EQUALRECT	Determines whether the two specified rectangles are equal by comparing their coordinates	Done	
245	ENABLECOMMNOTIFICATION	Enables or disables notification for a communications device	Stub	
246	EXITWINDOWSEXEC	Executes the specified application when Windows exits		
247	GETCURSOR	Retrieves the handle of the current cursor		
248	GETOPENCLIPBOARDWINDOW	Retrieves the handle of the window that currently has the clipboard open		
249	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		
250	GETMENUSTATE	Retrieves the menu item state	Done	
251	SENDDRIVERMESSAGE	Sends a message to an installable driver		
252	OPENDRIVER	Opens an installable driver		
253	CLOSEDRIVER	Closes an installable driver		
254	GETDRIVERMODULEHANDLE	Retrieves the module handle of an installable driver		
255	DEFDRIVERPROC	Provides default processing for any messages that an installable driver does not process		
256	GETDRIVERINFO	Retrieves information about an installable driver		
257	GETNEXTDRIVER	Retrieves the handle of the next installable driver in the list of installable drivers		
258	MAPWINDOWPOINTS	Converts (maps) a set of points from a coordinate space relative to one window to a coordinate space relative to another window		
259	BEGINDEFERWINDOWPOS	Allocates memory for a multiple-window-position structure and returns the handle to the structure		
260	DEFERWINDOWPOS	Updates the specified multiple-window-position structure for the specified window		
261	ENDDEFERWINDOWPOS	Simultaneously updates the position and size of one or more windows in a single screen-refreshing cycle		
262	GETWINDOW	Retrieves a handle to a window that has the specified relationship to the specified window		
263	GETMENUITEMCOUNT	Retrieves the number of items in the specified menu	Done	

Ordinal	Name	Description	Status	Version
264	GETMENUITEMID	Retrieves the menu item identifier of a menu item located at the specified position in a menu	Done	
265	SHOWOWNEDPOPUPS	Shows or hides all pop-up windows owned by the specified window		
266	SETMESSAGEQUEUE	Creates a new message queue for the calling application		
267	SHOWSCROLLBAR	Shows or hides the specified scroll bar		
268	GLOBALADDATOM	Adds a character string to the global atom table and returns a unique value (an atom) identifying the string	Done	
269	GLOBALDELETEATOM	Decrements the reference count of a global string atom, and if the reference count reaches zero, removes the string from the global atom table	Done	
270	GLOBALFINDATOM	Retrieves the atom associated with the specified character string from the global atom table	Done	
271	GLOBALGETATOMNAME	Retrieves a copy of the character string associated with the specified global atom	Done	
272	ISZOOMED	Determines whether a window is maximized		
277	GETDLGCTRLID	Retrieves the identifier of the specified control		
278	GETDESKTOPHWND	Retrieves the handle of the desktop window		
279	OldSetDeskPattern	Sets the desktop pattern	Done	
282	SELECTPALETTE	Selects a logical palette into a device context		
283	REALIZEPALETTE	Maps palette entries from the current logical palette to the system palette		
284	GETFREESYSTEMRESOURCES	Retrieves the percentage of free system resources	Done	
286	GETDESKTOPWINDOW	Retrieves the handle of the desktop window		
287	GETLASTACTIVEPOPUP	Determines which pop-up window owned by the specified window was most recently active		
288	GETMESSAGEEXTRAINFO	Retrieves the extra message information for the current thread		
290	REDRAWWINDOW	Updates the specified rectangle or region in a window's client area		
291	SETWINDOWSHOOKEX	Installs an application-defined hook procedure into a hook chain		
292	UNHOOKWINDOWSHOOKEX	Removes a hook procedure installed in a hook chain by the SetWindowsHookEx function		

Ordinal	Name	Description	Status	Version
293	CALLNEXTHOOKEX	Passes the hook information to the next hook procedure in the current hook chain		
294	LOCKWINDOWUPDATE	Disables or enables drawing in the specified window		
299	mouse_event	Synthesizes mouse motion and button clicks	Done	
308	DEFDLGPROC	Provides default processing for any messages that a dialog box with a private window class does not process		
309	GETCLIPCURSOR	Retrieves the cursor confinement rectangle		
319	SCROLLWINDOWEX	Scrolls the content of the specified window's client area		
324	FillWindow	Fills the client area of the specified window with the specified brush	Done	
325	PaintRect	Paints the specified rectangle by using the specified brush	Done	
326	GetControlBrush	Retrieves the brush for the specified control	Done	
331	ENABLEHARDWAREINPUT	Enables or disables mouse and keyboard input to the system		
333	IsUserIdle	Determines whether the system is idle	Done	
334	GETQUEUESTATUS	Retrieves the type of messages in the calling thread's message queue		
335	GETINPUTSTATE	Determines whether there are mouse-button, keyboard, or timer events in the message queue		
337	GetMouseEventProc	Retrieves the address of the current mouse event procedure	Done	
358	ISMENU	Determines whether a handle is a menu handle		
359	GETDCEX	Retrieves a handle to a device context for the client area of the specified window		
368	COPYICON	Copies the specified icon from another module to the current module	Done	
369	COPYCURSOR	Copies the specified cursor from another module to the current module	Done	
370	GETWINDOWPLACEMENT	Retrieves the show state and the restored, minimized, and maximized positions of the specified window		
371	SETWINDOWPLACEMENT	Sets the show state and the restored, minimized, and maximized positions of the specified window		
373	SUBTRACTRECT	Obtains the coordinates of a rectangle determined by subtracting one rectangle from another	Done	

Ordinal	Name	Description	Status	Version
397	REGISTERCLASSEX	Registers a window class for subsequent use in the CreateWindow or CreateWindowEx function	Done	
398	GetClassInfoEx	Retrieves information about a window class	Done	
402	GETPRIORITYCLIPBOARDFORMAT	Retrieves the first available clipboard format in the specified list		
403	UNREGISTERCLASS	Unregisters a window class, freeing the memory required for the class	Done	
404	GETCLASSINFO	Retrieves information about a window class	Done	
406	CREATECURSOR	Creates a cursor having the specified size, bit patterns, and hot spot	Done	
407	CREATEICON	Creates an icon having the specified size, colors, and bit patterns	Done	
408	CreateCursorIconIndirect	Creates an icon or cursor from an icon or cursor resource data structure	Done	
410	INSERTMENU	Inserts a new menu item into a menu, moving other items down	Done	
411	APPENDMENU	Appends a new item to the end of the specified menu	Done	
412	REMOVEMENU	Deletes a menu item or detaches a submenu from the specified menu	Done	
413	DELETEMENU	Deletes an item from the specified menu	Done	
414	MODIFYMENU	Changes an existing menu item	Done	
415	CREATEPOPUPMENU	Creates a drop-down menu, submenu, or shortcut menu	Done	
416	TRACKPOPUPMENU	Displays a shortcut menu at the specified location and tracks the selection of items on the shortcut menu	Done	
417	GETMENUCHECKMARKDIMENSIONS	Retrieves the dimensions of the default check mark bitmap	Done	
418	SETMENUITEMBITMAPS	Associates the specified check mark bitmaps with a menu item	Done	
420	_WSPRINTF	Writes formatted data to a string	Done	
421	WVSPRINTF	Writes formatted data to a string using a variable argument list	Done	
422	DLGDIRSELECTEX	Retrieves the current selection from a single-selection or multiple-selection list box filled by the DlgDirList function		
423	DLGDIRSELECTCOMBOBOXEX	Retrieves the current selection from a combo box filled by the DlgDirListComboBox function		
430	LSTRCMP	Compares two character strings	Done	
431	ANSIUPPER	Converts a character string or a single character to uppercase	Done	
432	ANSILOWER	Converts a character string or a single character to lowercase	Done	

Ordinal	Name	Description	Status	Version
433	ISCHARALPHA	Determines whether a character is an alphabetic character	Done	
434	ISCHARALPHANUMERIC	Determines whether a character is either an alphabetic or a numeric character	Done	
435	ISCHARUPPER	Determines whether a character is uppercase	Done	
436	ISCHARLOWER	Determines whether a character is lowercase	Done	
437	ANSIUPPERBUFF	Converts a character string to uppercase	Done	
438	ANSILOWERBUFF	Converts a character string to lowercase	Done	
445	DEFFRAMEPROC	Provides default processing for any window messages that a multiple-document interface (MDI) frame window does not process		
447	DEFMDICHILDPROC	Provides default processing for any window messages that an MDI child window does not process		
451	TRANSLATEMDISYSACCEL	Processes accelerator keystrokes for system menu commands of multiple-document interface (MDI) child windows		
452	CREATEWINDOWEX	Creates an overlapped, pop-up, or child window with an extended window style		
454	ADJUSTWINDOWRECTEX	Calculates the required size of the window rectangle based on the desired client rectangle size and window styles		
457	DESTROYICON	Destroys an icon and frees any memory the icon occupied	Done	
458	DESTROYCURSOR	Destroys a cursor and frees any memory the cursor occupied	Done	
462	CALCCHILDSCROLL	Calculates the scrolling rectangle for a scroll bar control		
466	DRAWFOCUSRECT	Draws a rectangle in the style used to indicate that the rectangle has the focus	Done	
471	LSTRCMPI	Compares two character strings, ignoring case	Done	
472	ANSINEXT	Returns a pointer to the next character in a string	Done	
473	ANSIPREV	Returns a pointer to the previous character in a string	Done	
482	ENABLESCROLLBAR	Enables or disables one or both arrows of a scroll bar		
483	SYSTEMPARAMETERSINFO	Retrieves or sets the value of one of the system-wide parameters	Done	
499	WNETERRORTEXT	Retrieves the error string for a network error	Stub	
501	WNETOPENJOB	Opens a print job on a network printer	Stub	

Ordinal	Name	Description	Status	Version
502	WNETCLOSEJOB	Closes a print job on a network printer	Stub	
503	WNETABORTJOB	Aborts a print job on a network printer	Stub	
504	WNETHOLDJOB	Holds a print job on a network printer	Stub	
505	WNETRELEASEJOB	Releases a held print job on a network printer	Stub	
506	WNETCANCELJOB	Cancels a print job on a network printer	Stub	
507	WNETSETJOBCOPIES	Sets the number of copies for a print job on a network printer	Stub	
508	WNETWATCHQUEUE	Installs a queue watch for a network print queue	Stub	
509	WNETUNWATCHQUEUE	Removes a queue watch for a network print queue	Stub	
510	WNETLOCKQUEUEDATA	Locks the queue data for a network print queue	Stub	
511	WNETUNLOCKQUEUEDATA	Unlocks the queue data for a network print queue	Stub	
512	WNETGETCONNECTION	Retrieves the network resource name associated with a local device		
513	WNETGETCAPS	Retrieves the capabilities of the network provider	Done	
514	WNETDEVICEMODE	Displays the configuration dialog box for a network device	Stub	
515	WNETBROWSEDIALOG	Displays a network browse dialog box	Stub	
516	WNETGETUSER	Retrieves the current network user name		
517	WNETADDCONNECTION	Connects a local device to a network resource	Stub	
518	WNETCANCELCONNECTION	Disconnects a network connection	Stub	
519	WNETGETERROR	Retrieves the most recent network error	Stub	
520	WNETGETERRORTEXT	Retrieves the error text for a network error	Stub	
521	WNETENABLE	Enables or disables the network provider		
522	WNETDISABLE	Disables the network provider		
523	WNETRESTORECONNECTION	Restores a network connection	Stub	
524	WNETWRITEJOB	Writes data to a network print job	Stub	
525	WNETCONNECTDIALOG	Displays a network connection dialog box	Stub	
526	WNETDISCONNECTDIALOG	Displays a network disconnection dialog box	Stub	
527	WNETCONNECTIONDIALOG	Displays a network connection dialog box	Stub	
528	WNETVIEWQUEUEDIALOG	Displays a network queue view dialog box	Stub	
529	WNETPROPERTYDIALOG	Displays a network property dialog box	Stub	
530	WNETGETDIRECTORYTYPE	Determines the type of a network directory		

Ordinal	Name	Description	Status	Version
531	WNETDIRECTORYNOTIFY	Notifies the network of a directory change	Stub	
532	WNETGETPROPERTYTEXT	Retrieves the property text for a network resource	Stub	