SendNotifyMessageA function (winuser.h) - Win32 apps

By jwmsft

Archived: 2026-04-05 19:59:11 UTC

Sends the specified message to a window or windows. If the window was created by the calling thread, **SendNotifyMessage** calls the window procedure for the window and does not return until the window procedure has processed the message. If the window was created by a different thread, **SendNotifyMessage** passes the message to the window procedure and returns immediately; it does not wait for the window procedure to finish processing the message.

Syntax

```
BOOL SendNotifyMessageA(
  [in] HWND   hWnd,
  [in] UINT   Msg,
  [in] WPARAM wParam,
  [in] LPARAM lParam
);
```

Parameters

`[in] hWnd`

Type: **HWND**

A handle to the window whose window procedure will receive the message. If this parameter is **HWND_BROADCAST** ((HWND)0xffff), the message is sent to all top-level windows in the system, including disabled or invisible unowned windows, overlapped windows, and pop-up windows; but the message is not sent to child windows.

`[in] Msg`

Type: **UINT**

The message to be sent.

For lists of the system-provided messages, see System-Defined Messages.

`[in] wParam`

Type: **WPARAM**

Additional message-specific information.

`[in] lParam`

Type: **LPARAM**

Additional message-specific information.

Return value

Type: **BOOL**

If the function succeeds, the return value is nonzero.

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

If you send a message in the range below WM_USER to the asynchronous message functions (PostMessage, **SendNotifyMessage**, and SendMessageCallback), its message parameters cannot include pointers. Otherwise, the operation will fail. The functions will return before the receiving thread has had a chance to process the message and the sender will free the memory before it is used.

Applications that need to communicate using **HWND_BROADCAST** should use the RegisterWindowMessage function to obtain a unique message for inter-application communication.

The system only does marshalling for system messages (those in the range 0 to (WM_USER-1)). To send other messages (those >= **WM_USER**) to another process, you must do custom marshalling.

Note

The winuser.h header defines SendNotifyMessage as an alias that automatically selects the ANSI or Unicode version of this function based on the definition of the UNICODE preprocessor constant. Mixing usage of the encoding-neutral alias with code that is not encoding-neutral can lead to mismatches that result in compilation or runtime errors. For more information, see Conventions for Function Prototypes.

Requirements

Requirement	Value
Minimum supported client	Windows 2000 Professional [desktop apps only]
Minimum supported server	Windows 2000 Server [desktop apps only]
Target Platform	Windows
Header	winuser.h (include Windows.h)
Library	User32.lib

Requirement	Value
DLL	User32.dll
API set	ext-ms-win-ntuser-message-l1-1-3 (introduced in Windows 10, version 10.0.14393)

See also

Conceptual

Messages and Message Queues

PostMessage

PostThreadMessage

Reference

RegisterWindowMessage

SendMessage

SendMessageCallback

SendNotifyMessage
