

ThreadProc callback function

An application-defined function that serves as the starting address for a thread. Specify this address when calling the [CreateThread](#), [CreateRemoteThread](#), or [CreateRemoteThreadEx](#) function.

The **LPTHREAD_START_ROUTINE** type defines a pointer to this callback function. **ThreadProc** is a placeholder for the application-defined function name.

Syntax

C++

```
DWORD WINAPI ThreadProc(  
    _In_ LPVOID lpParameter  
);
```

Parameters

lpParameter [in]
The thread data passed to the function using the *lpParameter* parameter of the [CreateThread](#), [CreateRemoteThread](#), or [CreateRemoteThreadEx](#) function.

Return value

The return value indicates the success or failure of this function. The return value should never be set to STILL_ACTIVE (259), as noted in [GetExitCodeThread](#).

Do not declare this callback function with a **void** return type and cast the function pointer to **LPTHREAD_START_ROUTINE** when creating the thread. Code that does this is common, but it can crash on 64-bit Windows.

Remarks

A process can determine when a thread it created has completed by using one of the [wait functions](#). It can also obtain the return value of its **ThreadProc** by calling the [GetExitCodeThread](#) function.

Each thread receives a unique copy of the local variables of this function. Any static or global variables are shared by all threads in the process. To provide unique data to each thread using a global index, use [thread local storage](#).

Examples

For an example, see [Creating Threads](#).

Requirements

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Header	WinBase.h on Windows XP, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008, and Windows Server 2008 R2 (include Windows.h); Processthreadsapi.h on Windows 8 and Windows Server 2012

See also

Community Additions
