WaitForSingleObject function

Waits until the specified object is in the signaled state or the time-out interval elapses.

To enter an alertable wait state, use the WaitForSingleObjectEx function. To wait for multiple objects, use WaitForMultipleObjects.

Syntax

C++

```
DWORD WINAPI WaitForSingleObject(
  _In_ HANDLE hHandle,
  _In_ DWORD  dwMilliseconds
);
```

Parameters

hHandle [in]

A handle to the object. For a list of the object types whose handles can be specified, see the following Remarks section.

If this handle is closed while the wait is still pending, the function's behavior is undefined.

The handle must have the **SYNCHRONIZE** access right. For more information, see Standard Access Rights.

dwMilliseconds [in]

The time-out interval, in milliseconds. If a nonzero value is specified, the function waits until the object is signaled or the interval elapses. If *dwMilliseconds* is zero, the function does not enter a wait state if the object is not signaled; it always returns immediately. If *dwMilliseconds* is **INFINITE**, the function will return only when the object is signaled.

Note The *dwMilliseconds* value does not include time spent in low-power states. For example, the timeout will not keep counting down while the computer is asleep.

Return value

If the function succeeds, the return value indicates the event that caused the function to return. It can be one of the following values.

Return code/value	Description
WAIT_ABANDONED 0x00000080L	The specified object is a mutex object that was not released by the thread that owned the mutex object before the owning thread terminated. Ownership of the mutex object is granted to the calling thread and the mutex state is set to nonsignaled. If the mutex was protecting persistent state information, you should check it for consistency.
WAIT_OBJECT_0 0x00000000L	The state of the specified object is signaled.
WAIT_TIMEOUT 0x00000102L	The time-out interval elapsed, and the object's state is nonsignaled.
WAIT_FAILED (DWORD)0xFFFFFFFF	The function has failed. To get extended error information, call GetLastError.

Remarks

The **WaitForSingleObject** function checks the current state of the specified object. If the object's state is nonsignaled, the calling thread enters the wait state until the object is signaled or the time-out interval elapses.

The function modifies the state of some types of synchronization objects. Modification occurs only for the object whose signaled state caused the function to return. For example, the count of a semaphore object is decreased by one.

The **WaitForSingleObject** function can wait for the following objects:

- Change notification
- Console input
- Event
- Memory resource notification
- Mutex
- Process
- Semaphore
- Thread
- Waitable timer

Use caution when calling the wait functions and code that directly or indirectly creates windows. If a thread creates any windows, it must process messages. Message broadcasts are sent to all windows in the system. A thread that uses a wait function with no time-out interval may cause the system to become deadlocked. Two examples of code that indirectly creates windows are DDE and the CoInitialize function. Therefore, if you have a thread that creates windows, use MsgWaitForMultipleObjects or MsgWaitForMultipleObjectsEx, rather than **WaitForSingleObject**.

Examples

For an example, see Using Mutex Objects.

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); Synchapi.h on Windows 8 and Windows Server 2012
Library	Kernel32.lib
DLL	Kernel32.dll

See also

Synchronization Functions
Wait Functions

Community Additions

Behavior when computers enter Suspend/Hibernation state

The documentation is actually wrong, the timeout keeps counting down while the computer is asleep/hibernated. Tested on Windows 7 SP1 amd64 system.


Hnanicek
6/12/2014

Missing Job object from list of objects this function can wait for

There is no job object in the following list: "The WaitForSingleObject function can wait for the following objects:"

but following page

"http://msdn.microsoft.com/en-us/library/windows/desktop/ms684161(v=vs.85).aspx"

contains statement that it is possible to use WaitForSingleObject for Job objects:

The state of a job object is set to signaled when all of its processes are terminated because the specified end-of-job time limit has been exceeded. Use WaitForSingleObject or WaitForSingleObjectEx to monitor the job object for this event.


Vlad Ch
7/13/2012

To make the Console wait for 't' seconds for an input

This program has a countdown timer. After the countdown gets over, the input session starts.

```
#include <time.h>
#include <conio.h>
#include <iostream>
#include <string>
#include <windows.h>
#define timer 5.0
using namespace std;
void main()
{
double dif=0;
time_t start,end;
time(&start);
string s;
while(dif<timer) //Countdown before starting
{
printf("Type a word after %.0lf seconds..\nYour Time starts in..\n",timer);
time (&end);
dif = difftime(end,start);
printf( "%.0lf seconds\n", timer-dif);
Sleep(1000);
system("CLS");
}
HANDLE h_to_console=GetStdHandle( STD_INPUT_HANDLE );
FlushConsoleInputBuffer(h_to_console);
cout<<"Enter the String:\n";
cin>>s;
cout<<"You have Entered: "<<s;
getch();
}
```

but, sorry this snippet is unrelated to WaitForSingleObject() anyway.

parthi
2/14/2012

Mutex behaviour, this should have been here too.

The thread that owns a mutex can specify the same mutex in repeated wait function calls without blocking its execution. Typically, you would not wait repeatedly for the same mutex, but this mechanism prevents a thread from deadlocking itself while waiting for a mutex that it already owns. However, to release its ownership, the thread must call ReleaseMutex ***once for each time*** that the mutex satisfied a wait.


pif75
5/26/2011

Current thread and process handles are special cased

Note that WaitForSingleObject (and WaitForSingleObjectEx) treats "special" handles (handles returned by GetCurrentProcess and GetCurrentThread) differently than *WaitForMultipleObjects* functions. The former functions return with WAIT_TIMEOUT, while the latter return that WAIT_FAILED, e.g.:

```
#include <stdio.h>
#include <Windows.h>

int main(int argc, char* argv[])
```

```
{
    HANDLE phandle = GetCurrentProcess();
    HANDLE thandle = GetCurrentThread();
    HANDLE ihandle = (HANDLE)-3;

    printf("%d %d %d\n", phandle, WaitForSingleObject(phandle, 0), WaitForMultipleObjects(1, &phandle, 0, 0));
    printf("%d %d %d\n", thandle, WaitForSingleObject(thandle, 0), WaitForMultipleObjects(1, &thandle, 0, 0));
    printf("%d %d %d\n", ihandle, WaitForSingleObject(ihandle, 0), WaitForMultipleObjects(1, &ihandle, 0, 0));
    return 0;
}
```

prints:

```
-1 258 -1
-2 258 -1
-3 -1 -1
```

So don't use WaitForSingleObject to determine which of the handles passed to *WaitForMultipleObjects* caused the wait to fail, use WaitForMultipleObjects with a single handle instead.

zseil
1/13/2011

Waiting for console input

Billy.Oneal, I believe you misunderstood. What Medinoc wanted was a function that could reliably tell him whether ReadConsole would block. On the surface, calling WaitForSingleObject on the hStdInput console handle does this, but what Medinoc found is that it serves that purpose only for individual characters, with the line input mode disabled. If you turn the line input mode on, then you end up with ReadConsole blocking even though WaitForSingleObject returns immediately with WAIT_OBJECT_0. I don't think it's unreasonable to want consistency between the functions, but it was evidently not part of the design of the console -- hence "not a bug". Still a design flaw.

Jonathan Gilbert
5/3/2010

Waiting for console input

Medinoc, that is not a bug. The "Dedicated, time limited wait function" is Sleep: http://msdn.microsoft.com/en-us/library/ms686298%28VS.85%29.aspx

This function is only for waiting on an object, not waiting a specified period.

Billy.Oneal
11/9/2009

C# syntax

```
[DllImport("kernel32", CharSet=CharSet.Ansi, SetLastError=true, ExactSpelling=true)]
internal static extern int WaitForSingleObject(IntPtr hHandle, int dwMilliseconds);
```

dmex
5/4/2009

VB.Net syntax

```
<DllImport("kernel32", CharSet:=CharSet.Ansi, SetLastError:=True, ExactSpelling:=True)> _
Public Shared Function WaitForSingleObject(ByVal hHandle As IntPtr, ByVal dwMilliseconds As Integer) As Integer
End Function
```

dmex
5/4/2009

Waiting for console input

Make sure to disable the ENABLE_LINE_INPUT console mode before waiting on a console input buffer object.

Regardless of the mode, console input buffer objects are signaled as soon as a character is typed, while the reading functions will block until the line is finished, voiding the purpose of a dedicated, time-limited wait function.

A Microsoft employee I contacted a few years ago about this issue claimed that it was not a bug.

Medinoc
11/4/2008

WaitForSingleObject Performance getting differed based on the OS used

Did WaitForSingleObject has more time delay when running in WindowsXP comparing with Windows2000?

[Noelle Mallory - MSFT] Please post questions to the MSDN Forums at **http://forums.microsoft.com/msdn**. You will likely get a quicker response through the forum than through the Community Content.

Noelle Mallory
11/26/2007