

Can I wait for a kernel event to become unsignaled?

 devblogs.microsoft.com/oldnewthing/20200305-00

March 5, 2020



Raymond Chen

There are various ways to wait for a kernel event to become signaled, such as `WaitForSingleObject`. Is there a way to wait for a kernel event to become *unsignaled*?

Not directly, but maybe you can fake it.

The traditional way is to create a pair of events, and manually keep them in opposite states. Whenever you set one event, you reset the other. One event represents the positive state, and the other represents the negative state, and people can choose the state they want to wait for.

Alternatively, if you aren't required to use kernel events, you can build your own synchronization objects that provide the desired functionality. For example, it's a simple exercise to take our [alternate manual-reset event built out of `WaitOnAddress`](#) and add support for waiting for the object to become unsignaled.

```
void WaitForAltManualEventToBeReset(ALT_MEVENT* Event)
{
    while (Event->State) {
        LONG Waiting = 1;
        WaitOnAddress(&Event->State,
                    &Waiting,
                    sizeof(Waiting),
                    INFINITE);
    }
}
```

A third option is to build your own synchronization object out of a condition variable. The condition variable has the advantage of permitting you to create arbitrarily complex wait conditions, so you can, for example, wait for all of a set of multiple event-like objects to become unset, or even something weird like “Wait for the number of set objects to be a prime number.”

[Raymond Chen](#)

Follow



