# If threads are created without a message queue, why can I post to them immediately upon creation?

devblogs.microsoft.com/oldnewthing/20241009-00

October 9, 2024


The documentation for Windows message queues says that

> To avoid the overhead of creating a message queue for non-GUI threads, all threads are created initially without a message queue. The system creates a thread-specific message queue only when the thread makes its first call to one of the specific user functions.

A customer was unable to observe the documented behavior. According to their experiments, they found that they were able to post a message to a thread immediately upon its creation!

```
DWORD CALLBACK ThreadProc(void* parameter)
{
    // This succeeds (?)
    BOOL success =
        PostThreadMessage(GetCurrentThreadId(), WM_APP, 314, 159);

    // So does this (!)
    MSG msg;
    success = PeekMessage(&msg, nullptr, 0, 0, PM_REMOVE);

    // And it's our message (!)
    if (msg.message == WM_APP) {
        OutputDebugStringW(L"We received the message (?)\r\n");
    }

    return 0;
}
```

What's going on here? We were able to post a message to the thread despite it never having created a message queue.

What's going on here is that `PostThreadMessage` is itself a message queue creation function!

So at the time you call `PostThreadMessage`, the thread does not have a message queue. When you call it, the system says, "First, I need to create a message queue for the current thread if it doesn't have one." This is what creates the message queue. And then we post the thread message into that queue.

If you want to see `PostThreadMessage` fail due to the lack of a message queue in the destination thread, then use it to post a message into the message queue of some *other* thread.

```
static auto ready = CreateEvent(nullptr, TRUE, FALSE, nullptr);
static auto exit = CreateEvent(nullptr, TRUE, FALSE, nullptr);

// Start a thread and wait for it to start.
DWORD id;
auto thread = CreateThread(nullptr, 0, [](void*) {
    SetEvent(ready);
    return WaitForSingleObject(exit, INFINITE);
}, nullptr, 0, &id);

WaitForSingleObject(ready, INFINITE);

// Now that it has started, try to post it a message.
auto success = PostThreadMessage(id, WM_APP, 314, 159);

// Tell the thread to exit.
SetEvent(exit);
```

In this case, the `PostThreadMessage` fails because the thread has not yet created a queue.

The customer's experiment failed because it made the thread post a thread message to itself, and the act of posting a thread message creates the message queue. To observe the absence of a message queue, you have to do the post from another thread.

So which functions create a message queue?

The functions that are guaranteed to create a message queue are `PeekMessage`, `GetMessage`, and `CreateWindow` (or other functions that create windows like `DialogBox`). There may be other functions that also create a message queue as a side effect (like `PostThreadMessage`), but you shouldn't rely on them.