

# If you suppress GDI+ background thread, then you are expected to pump messages yourself

[devblogs.microsoft.com/oldnewthing/20191029-00](https://devblogs.microsoft.com/oldnewthing/20191029-00)

October 29, 2019



Raymond Chen

When you initialize GDI+ by calling `GdiplusStartup`, you can choose to let GDI+ create its own notification thread (also known as the background thread), or you can suppress the notification thread and promise to do the work yourself. If you choose to do the work yourself, by setting `SuppressBackgroundThread` to `TRUE`, then the documentation says,

Call the hook and unhook functions before and after the application's main message loop; that is, a message loop that is active for the lifetime of GDI+.

In order to successfully put the `NotificationHook` function before your message loop and the `NotificationUnhook` function after it, you need to have a message loop to begin with.

The reason for the message loop is that GDI+ needs a helper window in order to listen for things like `WM_SETTINGSCHANGE` messages so it knows when to invalidate its caches or adjust its behavior depending on things like whether the session is on the console or is redirected via Remote Desktop. And that helper window needs a message loop so it can process messages.

Normally, this helper window goes on the notification thread, but you can assume responsibility for the helper window, and it will go onto *your* thread, which by virtue of now being a UI thread, must pump messages. The idea is that your application probably already has a UI thread, so you may as well add the notification window to that thread. That way, you save a thread.

Unfortunately, some people didn't quite understand this point. They saw the `SuppressBackgroundThread` option and said, "Hey, if I set this value to `TRUE`, then I save a thread!" They didn't read to the part where it says that you need to put a message loop on that thread.

The result is that the next time the system sends a message to the helper window, the helper window cannot receive the message because the thread isn't pumping messages, resulting in the usual negative consequences.

Next time, we'll look more closely at one of those negative consequences.

Raymond Chen

**Follow**

