

# What kind of messages can a message-only window receive?

 [devblogs.microsoft.com/oldnewthing/20171218-00](https://devblogs.microsoft.com/oldnewthing/20171218-00)

December 18, 2017



Raymond Chen

There's this thing called a *message-only window*, which is a very misleading name because receiving messages is the thing a window spends all its time doing anyway. It's like calling something a *water-only swimming pool* or a *heat-only oven*.

It's also a very misleading name because it *doesn't* receive all messages. "Wait, the name of this window says that the only thing it can do is receive messages, and now you're saying it can't do even that!"

The point of a message-only window is that it receives only messages sent or posted specifically to it. You use it to set up a private channel between the sender and the window. After creating a message-only window, you can put messages in the window's queue by calling `PostMessage` and passing that window handle, or you can send a non-queued message by calling `SendMessage` and passing that window handle.

What makes a message-only window interesting is that it doesn't participate in broadcast messages.

Many window messages are sent to all top-level windows. `WM_QUERYENDSESSION`, `WM_SETTINGCHANGE`, `WM_DDE_INITIATE`, and anything sent with `HWND_BROADCAST`. These messages don't reach message-only windows.

Internally, message-only windows are treated as child windows of a system-managed common parent window called `HWND_MESSAGE`. This system-managed common parent window is permanently invisible, which results in message-only windows being permanently invisible. And that's also how message-only windows are invisible to enumeration and broadcasts: Enumeration and broadcasting is done to top-level windows, but message-only windows are internally treated as child windows of `HWND_MESSAGE` and therefore are not considered top-level.

[Raymond Chen](#)

**Follow**

