What's so special about the desktop window?

devblogs.microsoft.com/oldnewthing/20040224-00

February 24, 2004

Raymond Chen

The window returned by GetDesktopWindow() is very special, and I see people abusing it all over the place. For example, many functions in the shell accept a window handle parameter to be used in case UI is needed. IShellFolder::EnumObjects, for example. What happens if you pass GetDesktopWindow()? If UI does indeed need to be displayed, you hang the system. Why?

- A modal dialog disables its owner.
- Every window is a descendant of the desktop.
- When a window is disabled, all its descendants are also disabled.

Put this together: If the owner of a modal dialog is the desktop, then the desktop becomes disabled, which disables all of its descendants. In other words, it disables every window in the system. Even the one you're trying to display! You also don't want to pass GetDesktopWindow() as your hwndParent. If you create a child window whose parent is GetDesktopWindow(), your window is now glued to the desktop window. If your window then calls something like MessageBox(), well that's a modal dialog, and then the rules above kick in and the desktop gets disabled and the machine is toast. So what window do you pass if you don't have a window? Pass NULL. To the window manager, a parent of NULL means "Create this window without an owner." To the shell, a UI window of NULL typically means "Do not display UI," which is likely what you wanted anyway.

Be careful, though: If your thread does have a top-level unowned window, then creating a second such window modally will create much havoc if the user switches to and interacts with the first window. If you have a window, then use it.

Raymond Chen

Follow