How do you troubleshoot WPF UI problems?

Asked 16 years, 3 months ago   Modified 15 years ago   Viewed 7k times

▲

11

▼

🔖

🕓

I'm working on a WPF application that sometimes exhibits odd problems and appears to *hang* in the UI. It is inconsistent, it happens in different pages, but it happens often enough that it is a big problem. I should mention that it is not a true hang as described below.

My first thought was that the animations of some buttons was the problem since they are used on most pages, but after removing them the hangs still occur, although seemingly a bit less often. I have tried to break into the debugger when the hang occurs; however there is never any code to view. No code of mine is running. I have also noticed that the "hang" is not complete. I have code that lets me drag the form around (it has no border or title) which continues to work. I also have my won close button which functions when I click it. Clicking on buttons appears to actually work as my code runs, but the UI simply never updates to show a new page.

I'm looking for any advice, tools or techniques to track down this odd problem, so if you have any thoughts at all, I will greatly appreciate it.

EDIT: It just happened again, so this time when I tried to break into the debugger I chose to "show disassembly". It brings me to MS.Win32.UnsafeNativeMethods.GetMessageW. The stack trace follows:

```
[Managed to Native Transition]
```

> WindowsBase.dll!MS.Win32.UnsafeNativeMethods.GetMessageW(ref System.Windows.Interop.MSG msg, System.Runtime.InteropServices.HandleRef hWnd, int uMsgFilterMin, int uMsgFilterMax) + 0x15 bytes
> WindowsBase.dll!System.Windows.Threading.Dispatcher.GetMessage(ref System.Windows.Interop.MSG msg, System.IntPtr hwnd, int minMessage, int maxMessage) + 0x48 bytes
> WindowsBase.dll!System.Windows.Threading.Dispatcher.PushFrameImpl(System.Windows.Threading.DispatcherFrame frame = {System.Windows.Threading.DispatcherFrame}) + 0x8b bytes
> WindowsBase.dll!System.Windows.Threading.Dispatcher.PushFrame(System.Windows.Threading.DispatcherFrame frame) + 0x49 bytes
> WindowsBase.dll!System.Windows.Threading.Dispatcher.Run() + 0x4c bytes
> PresentationFramework.dll!System.Windows.Application.RunDispatcher(obje

ct ignore) + 0x1e bytes

PresentationFramework.dll!System.Windows.Application.RunInternal(System.Windows.Window window) + 0x6f bytes

PresentationFramework.dll!System.Windows.Application.Run(System.Windows.Window window) + 0x26 bytes

PresentationFramework.dll!System.Windows.Application.Run() + 0x19 bytes

WinterGreen.exe!WinterGreen.App.Main() + 0x5e bytes C# [Native to Managed Transition]

[Managed to Native Transition]

mscorlib.dll!System.AppDomain.nExecuteAssembly(System.Reflection.Assembly assembly, string[] args) + 0x19 bytes

mscorlib.dll!System.Runtime.Hosting.ManifestRunner.Run(bool checkAptModel) + 0x6e bytes

mscorlib.dll!System.Runtime.Hosting.ManifestRunner.ExecuteAsAssembly() + 0x84 bytes

mscorlib.dll!System.Runtime.Hosting.ApplicationActivator.CreateInstance(System.ActivationContext activationContext, string[] activationCustomData) + 0x65 bytes

mscorlib.dll!System.Runtime.Hosting.ApplicationActivator.CreateInstance(System.ActivationContext activationContext) + 0xa bytes

mscorlib.dll!System.Activator.CreateInstance(System.ActivationContext activationContext) + 0x3e bytes

Microsoft.VisualStudio.HostingProcess.Utilities.dll!Microsoft.VisualStudio.HostingProcess.HostProc.RunUsersAssemblyDebugInZone() + 0x23 bytes

mscorlib.dll!System.Threading.ThreadHelper.ThreadStart_Context(object state) + 0x66 bytes

mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object state) + 0x6f bytes

mscorlib.dll!System.Threading.ThreadHelper.ThreadStart() + 0x44 bytes

wpf   debugging   user-interface   xaml

Share

Improve this question

Follow

edited Sep 19, 2008 at 18:26

asked Sep 19, 2008 at 18:17

palehorse

27.4k ● 4   ● 42   ● 49

5 Answers

Sorted by: Highest score (default) ⬍

6

Try removing the borderless behavior of your window and see if that helps. Also, are you BeginInvoke()'ing or Invoke()'ing any long running operations?

Another thing to look at: When you break into your code, try looking at threads other than your main thread. One of them may be blocking the UI thread.

Share Improve this answer Follow

answered Sep 19, 2008 at 18:21

Bob King
25.8k • 7 • 57 • 66

> There are times when some asynchronous process are executed; however this behavior can occur at any time, including right after startup when almost nothing has been run. I will try removing the borderless behavior. – palehorse Sep 19, 2008 at 18:23

> Just a note to other people reading this in the future: Never Invoke() or BeginInvoke() long-running operations on your Dispatcher. Your UI will appear to hang...use a thread or BackgroundWorker or something similar. – Bob King Sep 19, 2008 at 18:25

> Bob is right, if you're doing too much on the UI thread at one time the UI will become unresponsive. You want to do as little as possible on the UI thread; any big operations should be done on a background thread. – Rob Sep 19, 2008 at 18:28

6

Your WPF app could be hanging due to performance issues. Try using [Perforator](#) to see if you have any parts that are software rendered or if you app is using too much video ram.

Share Improve this answer Follow

answered Sep 19, 2008 at 18:26

Alan Le
8,823 • 7 • 37 • 31

> I did not know about that tool. I will be installing it shortly, – palehorse Sep 19, 2008 at 18:57

4

One great tool is [Snoop](#). Really nice for looking at what WPF objects are displayed on the visual tree at a given time. I'm not sure how much it will help, but it's possible you're jamming the UI thread with a lot of extra things it has to do. Snoop may be able to help you track down what is on the screen to give you an idea what to look for.

Share Improve this answer Follow

answered Sep 19, 2008 at 18:21

Rob
26.3k • 32 • 113 • 157

I have tried snoop, but as Bob noted, it does not help. I have noticed one other thing which I will edit the original question to include. – palehorse Sep 19, 2008 at 18:24

▲

4

▼

I have removed the borderless behavior as suggested by Bob King. To date, that seems to have gotten rid of the problem.

Now the question is, why and how can I fix the issue? The product is designed to have no border with some rounded corners and transparent parts.

Share Improve this answer Follow

answered Sep 22, 2008 at 17:12

palehorse
27.4k ● 4 ● 42 ● 49

I need this too. I've just had exactly the same issue. I noticed a MASSIVE performance gain when i switched to a windowed app too. – EightyOne Unite Dec 15, 2009 at 14:39

▲

1

▼

Hurrah,... it seems that the problem isn't related to borderless windows (at least in my case).

There is a big performance hit when you set `AllowsTransparency` to true. So much of a hit it would seem, that the whole thing can hang the UI thread. Very strange behaviour. Could be related to this ticket

Share Improve this answer Follow

answered Dec 15, 2009 at 15:41

EightyOne Unite
11.8k ● 14 ● 83 ● 107