

Re: Redirecting stdin, stdout, stderr from an already running process

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.interop/2008-03/msg00013.html>

- *From:* Jeroen Mostert <jmostert@xxxxxxxxxx>
 - *Date:* Mon, 03 Mar 2008 18:31:01 +0100
-

ghandi wrote:

On Mar 2, 5:59 am, Jeroen Mostert <jmost...@xxxxxxxxxx> wrote:

Jeroen Mostert wrote:

<snip>> A much simpler approach is to use impersonation, then use Process to

start the process regularly:

<snip>

This, finally, works on my system. Is it of use to you too?

Never mind, of course this doesn't work, and I'm a very sloppy tester.

Really, by this point, I can really only advise two courses of action:

- Give up and accept that there will be a window.
- After the process has started, use ugly code to find the window and hide it manually.

Doing it "properly" is way, way more trouble than its worth.

> Thanks again for your time. I'll try a few more things and then try

one of the things you mentioned (probably give up and accept that there will be a window). I was also looking at the new namespace System.IO.Pipes. It has an AnonymousPipeServerStream that might be useful.

If you don't mind the dependency on .NET 3.5 (since that's when they were introduced) then using the native pipe classes is definitely preferable to rolling your own. (I only recently discovered them myself, otherwise I would have recommended them.)

Re: Redirecting stdin, stdout, stderr from an already running process

—
J.

.