

Re: c# service unable to spawn exe

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.general/2006-10/msg00597.html>

- *From:* "Dr. Jochen Manns" <dev-2@xxxxxxxxxxxxxx>
 - *Date:* Thu, 19 Oct 2006 21:08:59 +0200
-

Im a bit upset because I'm running a service under WinXP SP2 and the following code works quite fine to spawn another process (in this case it's the service executable itself started as an isolated worker process communicating with the service by serializing commands using standard input and output):

```
// Create startup information
ProcessStartInfo pStart = new
ProcessStartInfo(Application.ExecutablePath, "AsChild");
```

```
// Use standard I/O
pStart.UseShellExecute = false;
pStart.RedirectStandardOutput = true;
pStart.RedirectStandardInput = true;
```

```
// Create new process
Process pNew = new Process();
```

```
// Attach
pNew.StartInfo = pStart;
```

```
// Activate it
pNew.Start();
```

Actually you should rethink the `CreateNoWindow = False` – maybe at least the interactive access to desktop would be required. But question is: do you really start the process with a GUI?

Jochen

"richie" <richardjneville@xxxxxxxxxxxxxx> wrote in message
<news:1161270595.351501.155030@xx>

Thanks for the reply Michael

I also contacted Microsoft and they said:

Re: c# service unable to spawn exe

System.Diagnostics.Process.Start is calling CreateProcessWithLogonW API under the scenes. The behavior of this API has changed in Windows XP SP2 and Windows Server 2003:

CreateProcessWithLogonW

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/createprocesswithlogonw.asp>

"

Windows XP SP2 and Windows Server 2003: You cannot call CreateProcessWithLogonW from a process that is running under the LocalSystem account, because the function uses the logon SID in the caller token, and the token for the LocalSystem account does not contain this SID. As an alternative, use the CreateProcessAsUser and LogonUser functions.

"

So basically starting a process from a c# service is not going to work when running under the local system account, which I need to do cos some users are restricted. I could use win32 via pinvoke to circumvent this.

But I've decided to write the service in win32 and run the code as a thread from the service. The exe I was running is c/c++ anyway so this is easy.

If anyone is interested a good article for writing a win32 service is <http://www.commssoft.com/services.html>

Thanks again

Cheers

Richard