

Cant create a explorer process with NT-AUTHORITY\SYSTEM Account,...

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2007-03/msg00068.ht>

- *From:* "Kerem Gümürkü" <kareem114@xxxxxxxxxxxx>
 - *Date:* Wed, 7 Mar 2007 09:05:52 +0100
-

Hi,

well this first sounds a bit stupid (and risky), but for some special reason i have to create a explorer.exe process with local system account Identity. The same task does work for a comand console but whenever i try to execute a explorer.exe it runs in my LoggenOn User Account (valid Pluto\Kerem Gümürkü for me). Why cant i start a explorer.exe with the Local System Account Token? The right question would be: Why does it even impersonate to my logged on user even when i start it with the Token from the Local System Account inside a (own, not shared) windows service process.

Scenario:

I wrote a C# User Interface that communicates with sockets and Custom Control Codes with the service. This works fine. I wrote two apps, a single one, pure C Windows API implementation and a pure C#.NET one. Both work with sockets and Service Control Codes and interact with the pure C native windows service. The bi-directional communication works fine. But the thing i dont understand is, why the console can be started with the NT-AUTHORITY\SYSTEM Account and the explorer.exe not? The explorer starts in the context of the service (service is flagged to be able to interact with the Desktop). Both Application start from the (own) Service Process. The Console runs with NT-AUTHORITY\SYSTEM but the explorer.exe with my logged on user token. Why?

Here is some (highly simplified) code:

```
/******  
if(OpenProcessToken(GetCurrentProcess(),TOKEN_ALL_ACCESS,&hToken) ==  
FALSE){  
  
char lpszErr[MAX_PATH];  
_ltoa((int)GetLastError(),lpszErr,10);  
OutputDebugString(lpszErr);  
return FALSE;
```

