

## Re: Webservice to an Out of process server

---

*Source:*

<http://www.derkeiler.com/Newsgroups/microsoft.public.inetsrv.iis.security/2006-11/msg00034.html>

---

- *From:* "news.microsoft.com" <[edousi@xxxxxxx](mailto:edousi@xxxxxxx)>
  - *Date:* Wed, 8 Nov 2006 15:53:05 -0000
- 

Hi Hankman,

I experience the same problem with the same setup.

I've create a COM server in VFP which I try to access from a Webservice using SOAP.

The webservice contains code:

```
loComServer = CREATEOBJECT('myComServer.app')
```

where myComServer.app is the class name as registered in the registry.

The only way I've been able to make it work is to adjust the Identity in the DCOM Config tool of the component services.

There seems to be a way of doing it using the IUSR\_XXX account (see text below) but I could not get that to work either.

Do you have got this working yet ?

From:

<http://blogs.msdn.com/david.wang/archive/2006/04/28/HOWTO-Run-Console-Applications-from-IIS6-on-Windows>

If you want to run executables on IIS from a script (i.e. an ASP, ASP.Net, or PHP page is considered a script resource executed by ASP.DLL, ASPNET\_ISAPI.DLL, or PHP-CGI.EXE / PHPISAPI.DLL Script Engine, respectively), then you need to configure "Scripts" execute permission as well as Web Service Extension for the appropriate Script Engine. i.e.

MyScript.asp contains the following content which executes FSUTIL.EXE:

```
<%  
set objShell = Server.CreateObject( "WScript.Shell" )  
objShell.Run( "FSUTIL.EXE" )  
%>1.. /cgi-bin has "Scripts" execute permission enabled.  
2.. %systemroot%\System32\inetsrv\ASP.DLL is enabled as a Web Service  
Extension.  
3.. /cgi-bin has a ScriptMaps property which associates .asp extension to  
%systemroot%\System32\inetsrv\ASP.DLL as a Script Engine.  
4.. You make a request to http://localhost/cgi-bin/MyScript.asp  
5.. IIS identifies ASP.DLL as the ISAPI Script Engine to process the
```

Re: Webservice to an Out of process server

/cgi-bin/MyScript.asp resource and checks it against Web Service Extension. Since it is allowed, it executes ASP.DLL using the user token obtained through whatever authentication protocol is negotiated between the browser and server.

Note: even though the ASP page runs FSUTIL.EXE, FSUTIL.EXE does NOT need to be in Web Service Extension because IIS never runs nor knows about FSUTIL.EXE. IIS only knows it is running ASP.DLL so that is what needs to be enabled as a Web Service Extension.

6.. ASP.DLL will keep the impersonated identity from IIS and parse/execute the script code in MyScript.asp using Windows Scripting Host. objShell.Run() translates into a CreateProcess() Win32 API call, and FSUTIL.EXE runs using the Process Identity (this is how CreateProcess is documented to work!)

7.. FSUTIL output is unknown to ASP (and IIS) unless you capture the output of objShell.Run() somehow and then Response.Write() it so that IIS knows about it.

Edz

"hankman" <hankman@xxxxxxxxxxx> wrote in message  
[news:12jerp21ut5cva4@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:12jerp21ut5cva4@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

WinXP Pro sp2  
IIS 5.1  
ASP.NET 2.0.50727  
Visual Studio 2005

I wrote a simple (I thought) webservice that talks to a running application that has an activeX interface.

Then I made a simple website to consume this web service. Everything works fine when running in the Visual Studio web development server which runs under my account permissions. But when I publish the web service to IIS the web service fails to obtain the running process object.

This leads me to believe that it is a security issue with IIS. So I made sure all the directories involved had admin privileges (NTSF) then added the IWAM and IUSR and even the Guest account to be admin's. Under IIS the Webservice as the Scripts and Executables permissions, but still no luck.

Are there any other security settings that I am missing?

Or is there some way to register the running process with IIS?

Thanks in advance.

Hank