Muse Vs. .NET Integration - Part 2

Posted At : October 19, 2010 11:20 AM | Posted By : Mark Kruger Related Categories: Coldfusion Troubleshooting

When last we left our story the Muse had put in a 6-7 hour marathon coding session through the night to create a CFC proxy work-around for his .NET problem. This temporary solution allowed the customer to continue working the following day so we could go back to scratching our heads over why .NET integration on our 64bit Multi-server was experiencing a failure to communicate. If you want to catch up on the story read my previous Part 1 post.

To recap, no matter what we tried we could not get CF to communicate with .NET on our production server (CF 8 Enterprise 64bit Multi-instance Install). We had tried installing and de-installing the service a few times to no avail. The first task was to get a successful recompile in 64bit that would run in a test environment. Team member (and all around genius developer) Guy Rish took charge of that effort. In fact, he got the assembly to recompile and he was able to instantiate and access its methods from his local CF8 64bit developer install running on Win7. From this we knew it was possible to get this working. We only needed to tease out the environmental differences to figure out what was blocking us. That task fell to me...

Where To Look?

The first problem I had was that it was hard to figure out exactly where to look. Most of what I needed was in the folder called "ColdFusion8DotNet". This folder is in the ColdFusion8 directory on a standard or Jrun/ent installation. In multi-server it is typically in the Jrun4 folder. If you have installed separately it will be wherever you chose to install it. Inside of this folder there are a couple of interesting files. One is "error.log" which records errors from .NET calls through the service. Another is the CF8DotNetsvc.log. This one shows errors from starting the .NET service.

The service log is the one that held the clue to what was going on here. Opening it up I found the following message "18/10/2010 14:15:01 INFO: StartProcess: Could not create process for 'C:\JRun4\ColdFusion8DotNet\JNBDotNetSide.exe', Error = 267". Hmmmm.... that sounds suspiciously like the service not being able to start. But I knew (or I thought I knew) that this was not the case. Just to be sure I opened the services control panel and restarted the "Coldfusion 8 .NET Service". It restarted quickly and with no complaint. Going back to the log I realized that the restart had created another identical error entry ("could not create process..."). So obviously Windows believed the service to be starting but internally the listener was not kicking off correctly.

Aha... Moment

Sometimes these processes reveal more information in console mode than they do in a log file. With that in mind I opened a command prompt and navigated to my coldfusion8dotnet folder. I typed in the name of the exe file that the log file was complaining about - JNBDotNetSide.exe - and hit enter. Nothing happened but the cursor did not return either. I suspected that I had either successfully launched the process or rendered the server useless. Naturally I was hoping for the former. I ran my test script and to my delight ColdFusion was able to instantiate and run my newly compiled .NET 64bit assembly. That's real progress!

Quick Recap

What had I learned? First, something about the *service* wrapper on ColdFusion Multi-Server was incorrectly configured. On a standard installation it worked fine - but something was different on multi-server that altered the way the *service started*. Just like Jrun the service for .NET integration depend on a "wrapper" called CF8DotNetsvc.exe. The purpose of this wrapper is to get the service running correctly without a shell and in the right security context etc. The real "listener" part of the process (the part that actually listens for Coldfusion to contact it) was actually the JNBDotNetSide.exe. So we needed to adjust something about the service parameters to get this fixed up.

Service Troubleshooting

That reminds me of when I was a child. I was sitting in church with my Mother (My Father was the preacher) and I noticed a wall of tiny plaques on the wall. I asked my mother the purpose of these plaques. She told me, "They are for all the men who died in the service." I pondered this for a moment and then asked "Which service did they die in, the 9:30 or 10:45?"

Anyway, my next step was to take a closer look at how windows was set up with this particular service. I knew from my test about that all my service wrapper really had to do was to kick off the listener. In fact, if I could not figure out how to fix the existing service I knew I could use a utility to simply install the listener as a new service (that was Plan B). Looking at the parameters in the services control panel I *immediately* saw something suspicious. In fact, it seemed like a smoking gun. The path to the CF8DotNetsvc.exe file was set to c:\Users*username*\tmp*some number*\\$JNBRIDGE_INSTALL_FOLDER\CF8DotNetsvc.exe. Surely that could not be the correct path.

I headed off to the registry to see what else was wonky with the service. I know there are lots of warning about working with the registry but it gives me a thrill - and it beats base jumping (less of a down side if you know what I mean). As registry risks go however, services are actually pretty uniform. I knew there was an "imagePath" that was pointed to the binary (the exe file) that was intended to run. So I needed to find that and re-point it to what I thought was the proper file (in

/Jrun4/ColdFusion8DotNet). What I found were 3 different settings that needed to be adjusted. Specifically I navigated to

"HKLM/System/CurrentControlSet/Services/Coldfusion 8 .NET Service" and altered 3 path settings:

- ImagePath
- CommandLine
- WorkingDir

In each case I removed the weird portion of the path that was there (which looked like the funky path referenced above) and replaced it with C:\Jrun4\Coldfusion8DotNet\. I restarted the service and checked the log file again. This time there was no error entry. I checked my test scripts and sure enough CF was now able to communicate with my .NET assembly.

Final Notes

A couple of other things we did that helped. We added the path to our assembly to the

JNBDotNetSide.exe.config file (in the /Coldfusion8DotNet/ folder). It has a node called "assemblyList" that looks like this.

```
<assemblyList>
<assembly file="C:\*somepath*\blah.dll" />
</assemblyList>
```

This sets up the assembly to be "bridged" to from CF - or so I gather. I had taken this to be more like a custom tag path, but apparently it's a cache config so ColdFusion can set up a Java stub for it. We also changed the .NET service permissions to "local system" to make sure it had full permissions to our assemblies and any native assemblies it referenced. I'm sure it would be possible to get it running as a permissioned account if needed.

Conclusion

So, if you are having trouble on ColdFusion enterprise Multi-server with .NET integration, check out the service settings and make sure they all point to the correct path for where your config files and service files are installed. Those of you with experience in this service feel free to comment. There is a dearth of tips or tricks for working with .NET Integration. I found little to go on over all.