# ColdFusion 8.01 64 bit on Windows 2k3 Web Edition - The Silent Killer

Posted At : February 24, 2010 2:49 PM | Posted By : Mark Kruger Related Categories: Coldfusion Troubleshooting

This post is a rundown of a recent troubleshooting adventure (If you don't know what a rundown is, ask Charles). I write these little post-mortems for a couple reasons. For one thing, I will need this solution (or some part of it) again at some point and my little pea brain is not going to retain all of this information. But what I usually do retain is the knowledge that somewhere on my blog I have a post about an issue with a Win2k3 64 bit CF install - or as Tevye would say, "Somewhere in the good book it must say something about a chicken!" The other reason is to benefit those of you in the ColdFusion universe who may be struggling with this issue. The Muse is all about community. If you are battling an install where the installer is failing silently after the splash screen, then this post will likely be a valuable read for you. If it doesn't solve your problem send me a note via Ask-A-Muse and I'll see if I can think of anything else you might try.

I was tasked with installing and optimizing ColdFusion 8 Enterprise 64 bit on a new server for a customer. I do this kind of task pretty often for companies who wish to have a clean and fully optimized install for migration. The new server was running Windows 2003 64 bit "Web Edition". Previously (in Windows 2000 I believe) the Web Edition would only support 2 Gigs of memory. But 2003 server supports up to 8 Gigs. I wanted to use 64 bit because the server was going to manage a large pool of modestly trafficked sites. It really didn't need the full fledged "multi-server" install. It just needed a nice big heap size. For that purpose the standalone Jrun install is perfect. In short order however, I found myself pitted against Win2k3 in a no holds barred install death match. Here's my story.

First things first. This customer has standardized on Apache - which is a very good choice. IIS performs well and is often easier to manage for windows sys admin types, a fact which often makes the Linux admin types (who think "GUI" is an adjective closely related to the noun "nougat") snicker. I have no objection to either and no stake in the argument. But the choice of Apache did present me with a problem. The official Apache site does not have 64 bit binaries for Apache. This is important because I didn't want to contend with a mixed environment. I wanted the Web server, OS and CF to all live in the same happy 64 bit space. Luckily - like most well supported open source projects - I found the recompiled binaries for 2.2 from another vendor, tested and installed them. Problem 1 solved. On to the install.

I opened my freshly downloaded 64 bit install file and "installanywhere" started unpacking the files. Eventually I was greeted with the main splash screen for the install. You know the one. It has a little "English" (or whatever language) drop down box and an "OK" button at the very bottom. I clicked "OK" and waited. The progress bar went up to about 80% and the splash screen disappeared to be replaced with ... nothing. In the words of Borimir, "What is this new devilry?" (and Sean Bean says it so well).

Over the next 10 or 15 minutes I tried various things.

- Downloaded a new install file.
- Used a "known good" install file.

- Turned off DEP with using /noexecute=AlwaysOff in the boot INI file.
- Cleared out the TEMP directory on the off chance that there was a locked file.
- Changed the TEMP directory to an empty folder with the "Everyone" group given full access.

None of this worked and I was left scratching my head.

## **Finding some Clues**

Having exhausted all the "easy" ideas I proceeded to dig around in the temp directory looking for clues. Actually, I opened the temp directory, emptied it, then clicked on the install file again and watched. The install file created a "1" directory in which it placed another directory (Ixxxxx) which contained 2 folders called "installerData" and "windows" and a funky file called sea\_loc.



Before clicking ok on the little button I copied these 3 objects to a new folder. Then I clicked "ok". The error crapped out as before.

Ignoring the temp directory I went to the new location where I had stored the files created by the installanywhere process. I surmised (correctly) that these were all the install files "unpacked" and that I could likely start the install using a file contained in one of the folders. The "installerdata" folder had a couple of archives and manifests in it, but the "Windows" folder had a file titled "adobe\_cf.exe". It also had a file called "cfinstall\_err". I opened this file and found that it contained a summary of system stats and variables that the installer needed, but there were no errors - at least nothing I could hang my hat on. The file contained Java configuration information (version, class path, temp directory etc) - but it really resembled a config file - or perhaps a standard output file. It did not appear to be an error log to me.



I decided to execute the adobe\_cf.exe file. Sure enough it displayed the "splash" screen. I clicked "OK" like before and the install failed silently - just like before. But this time I was watching the folder. I noticed that 3 files were written to it right away. the was an xxxxxx.notes file (a cryptic string as a file name) an xxxxx.details file and a file I was all too familiar with called hs\_err\_pidXXXX.log. What's this? A hotspot compiler error? I'm used to seeing these on sick servers in the /runtime/bin directory - but on this server Java was throwing me a hotspot error during the install.

The file contained mostly memory registers and I found little usefulness in anything past the header. Here's what the header looked like.

```
# An unexpected error has been detected by Java Runtime Environment:
  EXCEPTION ACCESS VIOLATION (0xc0000005) at pc=0x0000000077ef65ca, pid=3020, tid=3836
#
#
#
 Java VM: Java HotSpot(TM) 64-Bit Server VM (10.0-b19 mixed mode windows-amd64)
 Problematic frame:
#
 C [ntdll.dll+0x365ca]
#
 If you would like to submit a bug report, please visit:
#
 http://java.sun.com/webapps/bugreport/crash.jsp
The crash happened outside the Java Virtual Machine in native code.
#
#
 See problematic frame for where to report the bug.
#
#
.... later on...
Java version 1.6.0 04
```

Not much to go on, but at least I had a place to start. I did some googling and found no references to this specific error (win2k3 + 64 bit + ntdll.dll). I did find some version compatibility notes relating to JNI and a 64 bit JVM. That got me thinking about he JVM version. Hmmm I thought... whatever JVM the installer is using isn't playing nice with this version of windows. In fact, one fairly common fix for a hotspot error is to upgrade or downgrade the JVM (depending on your specific issue). So, thought I, I will install a different JVM and see if the system picks it up from the environment (there was a path clue in the HS file as well - pointing to a Java file buried deep in the

#### Windows directory.

I downloaded and installed the latest and greatest JVM (1.6.0\_18) and tried again. Not only did it produce the same error, but it indicated the same old Java version (1.6.0\_04) nothing had changed apparently. How do I get this installer to use a different JVM to go through it's paces?

## Aha Moment

Going back to the folder I noticed a file called "adobe\_cf.lax". Eureka! This file contains a good number of settings that seem to be environmental setup for the JVM. Here's a small snippet:

lax.stdout.redirect=cfinstall\_err .... lax.user.dir=. .... lax.stderr.redirect=cfinstall\_err .... lax.resource.dir=Windows ... lax.nl.java.option.java.heap.size.initial=16777216 .... lax.nl.current.vm=resource\\jre\\bin\\java.exe

Hmmm... standard out and standard error both directed to that "cfinstall\_err" file. And look at that - I have a specific JRE directory. Ignoring the warning at the top of the page to "not make any changes to this file" I edited the "current.vm" to point to the new 64 bit Java.exe I had just downloaded. My new string looked like this:

lax.nl.current.vm=c:\\jdk1.6.0\_18\\jre\\bin\\java.exe

When I fired up the install and clicked on my "ok" button one more time, sure enough the install launched and completed without incident. I ran the Apache connector script, located the CFIDE Admin, set up my new JVM settings and request settings and I was off and running.

# An Additional Insight

CFG Matt Woodward suggested I try a "silent" install (instructions here). This is actually a pretty easy process that involves building a properties file and specifying it as an argument to the install file (launching the install file from the command line). I did not have an adequate test of that idea and (now that the problem is solved) I have no test machine currently throwing a hotspot error on install. If the issue is that he *GUI features of the JVM that ships with CF don't cooperate well with the version of windows I was using*, then I suspect a silent install will probably work. Hopefully an intrepid reader can try that approach first and give us an update before using my work around solution.