

Herein lies the complete instructions for getting a Windows Server 2012 R2 x64 WSUS Server to install along side an IIS Application that requires a 32-bit application pool... I needed to do this because the IIS application I use on the WSUS Server uses a Classic ASP Application that has a 32-bit ODBC driver for Sybase (v16). My application really just queries databases and displays results.. but one of the applications I needed to look at was on SyBase...

Who the heck uses Sybase anymore? Apparently people still do... ☺

When I had the application pool configured to run in 64 bit mode, this is the error I received.

[Microsoft][ODBC Driver Manager] The specified DSN contains an architecture mismatch between the Driver and Application. ADO Returned 0x80004005

After changing the application pool to 32-bit mode, my application was gone and replaced by a not-so-friendly IIS 8.5 error page with an error 500.19 and a 0x800000c1

My error is pretty much dead-on for the error on this website... Kudos to the people that wrote this up... Although I couldn't find a way to add a precondition to the scheme mapping (as they suggested), but I am not exactly an ASP.NET guru.

<https://blogs.msdn.microsoft.com/webtopics/2010/03/08/troubleshooting-http-500-19-errors-in-iis-7/>

A lot of the information for this write up came from varied resources on the net, but after searching for about a day and only finding references to a few places that suggested we remove the compression driver and not worry about the consequences (<https://forums.iis.net/t/1149768.aspx>), these 2 websites pretty much gave me all I needed to figure out the total solution (And present it to you).

- <https://conetrix.com/Blog/internal-server-errors-on-64-bit-wsus-with-other-32-bit-websites-in-iis>
  - This site was helpful as well in understanding the true issue at hand, and the fact that we were really just missing a DLL... THANKS MICROSOFT!!!!
- <https://forums.iis.net/t/1149768.aspx>
  - ...this helped me understand how to fix the problem (providing I could find an x86 version of suscomp.dll)
- <https://blog.lextudio.com/2010/06/iis-7-and-wsus-on-windows-server-x64/>
  - ....but this had helpful notes here that helped me to complete solution and how to get suscomp.dll for x86 (although not exactly a complete solution..... just a blurb on the filename) :)

(Tested in Windows Server 2012 R2 x64 with WSUS 6.3.9600.18228)

\*\*\*\*Yes, I used WinRAR for part of this and 7Zip for another, with good reasons -- neither product was totally capable of extracting the EXE, MSI and CAB files 100% correct – that I could find, in limited testing). You may get stuff to work correctly using another extraction program, but this is how I did it.

Applications used:

WinRar 5.21 (32 bit)

7Zip 15.1 (x64)

On The system you are downloading the files from (possibly a Windows 10 PC):

1. Download WSUS SP2 x86 (WSUS30-KB972455-x86.exe) using your favorite browser  
<https://www.microsoft.com/en-us/download/confirmation.aspx?id=5216>
2. Extract exe to directory using →**WinRar**← to a subdirectory
  - Once extracted to there, the extracted files/directories should have a SUS directory... change directories to there...
3. Extract the **WUSSetup.msi** using →**7Zip**← to a sub directory
  - you are now looking for a file named MSUSCAB... copy it somewhere and rename it to MSUS.CAB (or edit the file name in that directory) so you can use win32 apps on your computer for extracting CAB files
4. Browse that cab or extract all the files from it using →**WinRar**← to a subdirectory, and find this file: (it should be the 9<sup>th</sup> file in the directory listing, and the 2<sup>nd</sup> file to start with "f\_" if your windows explorer is sorted by file name – it is approximately 22kb in size):
  - **f\_0DBF0D09\_4000\_43D6\_BA6D\_7226BD3BE5B8**
5. Rename the file you found in the previous step to **suscomp.dll** and copy it to your server:

On the 2012 IIS Server :

1. Copy the files to their new destinations
  - copy the x86 version of **suscomp.dll** from the previous step to %windir%\syswow64\inetsrv
  - copy C:\Program Files\Update Services\WebServices\suscomp.dll to %windir%\system32\inetsrv

(It's possible you might get it to work at this point, but I couldn't... so, I deleted and recreated the mapping as below)

2. Open a command prompt with Administrative privileges... (find cmd.exe and "Run as Administrator")
3. If you are worried about the config, backup the config using appcmd:
  - Here's a great wiki-style webpage on how to do the backups and restores... Yes, it still works for IIS 8.5

- <http://blogs.iis.net/bills/how-to-backup-restore-iis7-configuration>
4. delete the existing compression scheme for WSUS:
    - %windir%\system32\inetsrv\appcmd.exe set config - section:system.webServer/httpCompression /-[name='xpress']
  5. Re-add the xpress compression scheme (for WSUS) so it looks for it in "C:\Windows\SysWow64\inetsrv" & "Windows\System32\inetsrv" directories
    - %windir%\system32\inetsrv\appcmd.exe set config - section:system.webServer/httpCompression /+[name='xpress',doStaticCompression='false',dll='%windir%\system32\inetsrv\suscomp.dll']
  6. Restart IIS
    - Run iisreset.exe from a command prompt
  7. Validate the WSUS console works (and monitor the SoftwareDistribution.log and Event Viewer)
    - C:\Program Files\Update Services\LogFiles\SoftwareDistribution.log
    - Server Manager -> Tools -> Event Viewer -> Windows Logs -> Application
  8. Configure your 32-bit app pools for the website you need working again with 32-bit application pools (NOT WSUS)
    - Go to the "Internet Information Services (IIS) Manager" and click on: ServerName->Application Pools
    - On the Right Hand side of this console, click "Advanced Settings" and then change "Enable 32-Bit Applications" to "True" and Click OK.

\*\*\*While I can't guarantee this will not get "borked" up at some point by a Windows Update, this will now work to dual home 32-bit Application pools along side 64-bit application pools on a WSUS server since Microsoft didn't care to make this work by default. And considering you probably won't ever invoke the 32-bit version of this http compression scheme, security considerations shouldn't be an issue there either.

Steven Manross

2016/05/29