Ruby master - Bug #12371

Windows Nano Server WIN32OLE compatibility

05/11/2016 05:58 PM - alexpilotti (Alessandro Pilotti)

Status:	Closed		
Priority:	Normal		
Assignee:	suke (Masaki Suketa)		
Target version:			
ruby -v:		**Backport:**	2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description

The OleInitialize() call used in WIN32OLE is not supported on Nano Server due to the fact that the STA COM model is not available:

https://github.com/ruby/ruby/blob/32674b167bddc0d737c38f84722986b0f228b44b/ext/win32ole/win32ole.c#L820

As an alternative, CoInitializeEx(NULL, COINIT_MULTITHREADED) can be called when running on Nano Server, thus ensuring both compatibility with the preexisting behaviour and support for Nano Server.

History

#1 - 05/12/2016 01:58 AM - nobu (Nobuyoshi Nakada)

- Description updated

- Status changed from Open to Assigned

- Assignee set to suke (Masaki Suketa)

Doesn't CoInitialize differ than OleInitialize?

#2 - 05/12/2016 04:20 PM - Iristyle (Ethan Brown)

Yes, the behavior of OleInitialize is generally additive and per the docs at https://msdn.microsoft.com/en-us/library/windows/desktop/ms690134(v=vs.85).aspx, it is necessary to call before using any COM functions that use:

- Clipboard
- Drag and Drop
- Object linking and embedding (OLE)
- In-place activation

Under the hood, OleIntialize calls CoInitializeEx with COINIT_APARTMENTTHREADED, but that value is unsupported on Nano. See the COINIT enumeration docs for more details at https://msdn.microsoft.com/en-us/library/windows/desktop/ms678505(v=vs.85).aspx

I think the main point here is that calling OleInitialize will never work on Nano Server, since Nano only supports COINIT_MULTITHREADED, which OleInitialize cannot specify. Nano does support COM, but not all of the functionality that Ruby might normally enable on other platforms by calling OleInitialize. It's better in this case to have a degraded experience on Nano that still allows WIN32OLE to work / interact with COM objects, even if it doesn't support OleInitialize. Also note that WIN32OLE.connect will not work since Nano does not support COM monikers.

#3 - 05/12/2016 09:21 PM - cremno (cremno phobia)

In my tests OleInitialize(NULL) succeeded on Nano Server TP5 and reverse forwarders installed. Then it already is basically equivalent to CoInitializeEx(NULL, COINIT_MULTITHREADED) (everything else it does is printing two debugging messages). Does it fail for you?

#4 - 05/13/2016 03:52 AM - Iristyle (Ethan Brown)

In Window Nano Server TP5, Microsoft has temporarily changed the behavior of OleInitialize internally to use CoInitializeEx(NULL, COINIT_MULTITHREADED) as a compatibility fix, despite what the spec / documentation says. It is understood that this will not be the final shipping behavior. I have no information on the timeline, but you can see some more notes at https://github.com/puppetlabs/facter/pull/1327/files#diff-6c0a571515f6f937e35e5b86cbeb8821R11

#5 - 08/30/2016 05:10 PM - Iristyle (Ethan Brown)

Masaki - have you had a chance to look at this? The Nano Server RTM is due out soon, and it will fail when OleInitialize is called.

At that point, the Ruby WIN32OLE support will cease to work on Nano.

Anything that can be done to address that prior to it failing in the wild would be great.

Thanks!

#6 - 08/31/2016 11:58 AM - suke (Masaki Suketa)

Ethan, It's not easy for me to use CoInitializeEx instead of OleInitialize.
I had tried to use CoInitializeEx(NULL, MULTITHREADED) before, but I encountered SEGV about thread issue
and could not fix it.
To make Ruby's thread and win32ole's thread work well, I use OleInitialize and CoRegisterMessagefilter.
Unfortunately, CoRegisterMessagefilter works with OleInitialize.
I'll try to investigate this issue again, but I'm not sure to fix it.
Do you have any good idea to fix this issue?

#7 - 01/30/2017 10:23 PM - Iristyle (Ethan Brown)

Masaki -

I apologize for missing your response. There is a pull request open at https://github.com/ruby/ruby/pull/1423 from one of the engineers at Chef addressing this issue. If you could review, that would be great.

Thanks!

#8 - 10/21/2019 04:36 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Assigned to Closed

This appears to be fixed by 8feb9779182bd4285f3881029fe850dac188c1ac.