

Runtime Infrastructure - Feature #3268

Creating a tool to install and manage FWD server as Windows service

03/16/2017 02:15 PM - Eugenie Lyzenko

Status:	New	Start date:	03/16/2017
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to User Interface - Bug #3250: win_spawn failure cannot create process			Closed

History

#1 - 03/16/2017 02:27 PM - Eugenie Lyzenko

The objective is to implement Windows application that can be used to install CMD server starting script(or equivalent) to launch the server as Windows service.

The currently existed applications like:

<http://www.rozanski.org.uk/software>

<http://nssm.cc>

do not fully meet our needs.

Here is the list of some links can be used as starting point for such development:

<https://msdn.microsoft.com/en-us/library/bb540476%28v=VS.85%29.aspx>

<https://code.msdn.microsoft.com/windowsapps/CppWindowsService-cacf4948>

<http://www.devx.com/cplus/Article/9857>

The service running condition is to be executed with FWD application owned(user fwd) and be accessed via web client with any other user that is valid to run the web client. In terms of the service there is single owner and multiple clients capable to use the service.

#2 - 03/16/2017 11:38 PM - Eugenie Lyzenko

The current approach we use while missing our own service utility is to use nssm as service. After copy nssm.exe to the Windows system32 directory the install command:

```
nssm install FwdAppServer c:\opt\poc\deploy\server\server.cmd
```

Checking the installed service:

```
nssm edit FwdAppServer
```

The key point here is to verify the working directory is the same the server.cmd is located. Also the service owner must be the user known as server manager(fwd).

From this time the server can be started either from control panel or manually by:

```
sc start FwdAppServer
```

The main issue is the web client is working OK only for user that is owner of the service. On Windows 7 the attempt to access by another user causes the Access Denied error in CreateProcessWithLogonW. In Windows XP it causes the java.exe to crash.

The resolution is to change the service descriptor settings to allow required rights for Authenticated Users(...;AU):

```
sc sdset FwdAppServer D:(A;;CCLCSWLOCRRCGAF;;;AU)(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCLCSWLOCRRC;;;IU)(A;;CCLCSWLOCRRC;;;SU)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)
```

This can help with access denied issue but not a solution because discovers new issue with java.exe process silently ending.

Here is the scenario:

1. The server service is running as evl user. The service descriptor is:

```
sc sdset FwdAppServer
```

```
D:(A;;CCLCSWLOCRRCGAF;;;AU)(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCLCSWLOCRRC;;;IU)(A;;CCLCSWLOCRRC;;;SU)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)
```

meaning the Authenticated Users(...;AU) has GA and FA to execute and file access.

2. Starting the Firefox, login to localhost:7443/gui using account other than service user.

3. The spawn.exe executing and CreateProcessWithLogonW called without error, but nothing is happening next. No records in server.log. client.log is not even creating(the working directory is referring to proper client dir).

4. After some time the browser displays: "Cannot start embedded server."

Looks like need to debug more into java process started by spawn.exe to find out how it can silently disappear. And this is very strange. What we know for sure is the spawn.exe executing and calls CreateProcessWithLogonW without error.

#3 - 03/16/2017 11:41 PM - Eugenie Lyzenko

The Wiki document section for automated server starting was also updated with recent changes for use NSSM.EXE.

#4 - 03/17/2017 08:51 PM - Eugenie Lyzenko

Another info from mail conversation:

Here is the running FWD server application as service in Windows status. In short the result is not as good as expected.

When running the FWD server as fwd user the web client is able to log on and start the application. But when trying to access the web client via another user (evl for example) the web client is not starting. The spawn.exe is executing and failure is in CreateProcessWithLogonW, with message "Access Denied". The server is running as fwd user. Interesting point is if the server is running as fwd in application mode - it is possible to run web client with user other than fwd.

So for some reasons the service context does not allow to make CreateProcessWithLogonW call. I think it is all about security issue.

The plan is:

1. Investigate how we can set up service security context to allow arbitrary user to log on properly.
2. Investigate whether we can start the server in application mode. For example to change the cmd to initiate server.cmd running as application.

#5 - 03/17/2017 09:49 PM - Eugenie Lyzenko

The today investigation results.

The new are pretty sad. The shortly speaking the CreateProcessWithLogonW can not be called from service context for the user other than service owner. It looks like it is called without error but this is not a true. The java application ends abnormally with error 0xc0000142 - application failed to initialize properly. The system thinks the java.exe file or other required components are corrupted. In Windows 7 this is not seen on the screen unlike XP but can be found in error logger.

The most informative explanation I've found digging the web is here (not directly related to C code but idea is the same):

<http://www.asprosys.com/Resources/omphalos/perilsandpitfalls.aspx>

The my understanding is the service owner log on to the system in service security context and uses the (window station)\desktop other than default "winsta0\default" in service run mode and because the owner has full rights for this object - the process is starting and web client is running. When starting spawn.exe with another user the process needs access to the desktop not having enough rights.

So for now my conclusion is we need to re-write the winspawn code to be compatible with service mode or may be run every client with service owner logon (using CreateProcess() - this works from service).

The interesting and a bit weird fact is even to make another user to act as system service - the failure does not change, not possible to properly call into CreateProcessWithLogonW.

I need to think about what we can do here. As I understand the running all web clients from single service starter logon is not acceptable and not solution, right? Do we have a time to experiment with winspawn.c changes to be compatible with service mode?

Additionally I thought about to make a small stub for service command file like start /c server.cmd to run application from service in separate session. But even if this will work this will be not a service because does not have the ability to stop/restart the server.

#6 - 03/20/2017 10:59 AM - Greg Shah

- Related to Bug #3250: win_spawn failure cannot create process added

#7 - 03/20/2017 11:04 AM - Greg Shah

Can the FWD server be run properly for normal usage (access from a FWD client that is manually started)?

Do I understand correctly that it is only the spawning that is broken in windows service mode?

BTW, I wonder if we could run the remote spawner as a service. Trying to run the local spawning is probably not a good idea from a service. But the remote spawner may work. Still, it may have the same limitation of not being able to access the right desktop.

#8 - 03/20/2017 11:23 AM - Eugenie Lyzenko

Greg Shah wrote:

Can the FWD server be run properly for normal usage (access from a FWD client that is manually started)?

Yes, if the server is starting manually after user logon to the Windows session.

Do I understand correctly that it is only the spawning that is broken in windows service mode?

For both cases:

1. Running server.cmd as service in XP and Windows 7 and
2. Running server.cmd from Windows 7 Task Scheduler.

Yes the problem is in win_spawn.c code with CreateProcessWithLogon() or in environment block creation a bit before.

#9 - 03/20/2017 11:26 AM - Greg Shah

Can the FWD server be run properly for normal usage (access from a FWD client that is manually started)?

Yes, if the server is starting manually after user logon to the Windows session.

So nothing works at all if you start the server as a service? You can't connect to that server using a FWD client on a different machine?

That is surprising, since there is no process launching in that case.

Bor

What is "Bor"?

#10 - 03/20/2017 11:38 AM - Eugenie Lyzenko

Greg Shah wrote:

Can the FWD server be run properly for normal usage (access from a FWD client that is manually started)?

Yes, if the server is starting manually after user logon to the Windows session.

So nothing works at all if you start the server as a service? You can't connect to that server using a FWD client on a different machine?

That is surprising, since there is no process launching in that case.

If we start server as service it is possible to run web client for user that is the same as service owner. Using all other users in the web client login page causes the web client to crash with 0xC0000142.

Bor

What is "Bor"?

Sorry, I should be more accurate writing a notes. Fixed.

#11 - 03/20/2017 11:47 AM - Greg Shah

If we start server as service it is possible to run web client for user that is the same as service owner. Using all other users in the web client login page causes the web client to crash with 0xC0000142.

I'm not asking about the web client. I want to know if you can connect to a FWD server running as a windows service, using a completely separate FWD client. For example, a FWD client running on a Linux system. Does the FWD server work normally for this case?

#12 - 03/20/2017 12:14 PM - Eugenie Lyzenko

Greg Shah wrote:

If we start server as service it is possible to run web client for user that is the same as service owner. Using all other users in the web client login page causes the web client to crash with 0xC0000142.

I'm not asking about the web client. I want to know if you can connect to a FWD server running as a windows service, using a completely separate FWD client. For example, a FWD client running on a Linux system. Does the FWD server work normally for this case?

No, in this case FWD server does not work normally for any user(including service owner).

#13 - 03/20/2017 12:34 PM - Greg Shah

I don't understand. The FWD client on Linux cannot connect to the FWD server on Windows? What is the failure? In this example, there is no spawning. So if the FWD server starts and can open its TCP/IP ports, then it should work fine.

#14 - 03/20/2017 12:44 PM - Eugenie Lyzenko

Greg Shah wrote:

I don't understand. The FWD client on Linux cannot connect to the FWD server on Windows? What is the failure? In this example, there is no spawning. So if the FWD server starts and can open its TCP/IP ports, then it should work fine.

The FWD client on Linux can connect to the FWD server running on Windows. But after user enters the login data on the Linux side and waits for the client to start -> the server executing spawn.exe on Windows which in turn starts the java.exe on Windows to provide data for Linux side browser. And process started from spawn.exe on Windows is not working properly causing the message "Can not start embedded server" on the Linux side browser.

#15 - 03/20/2017 01:26 PM - Greg Shah

I know the web client doesn't work. Use the Swing client on Linux and connect to the Windows server running as a service. Does that work?

#16 - 03/20/2017 02:30 PM - Eugenie Lyzenko

Greg Shah wrote:

I know the web client doesn't work. Use the Swing client on Linux and connect to the Windows server running as a service. Does that work?

I'm in shock. It works. No matter if the Linux user name is the same as service owner name. It works for every client's user case. The server is running as service. Looks like a good news.

Do I need to check another Windows machine as client to connect Windows server with Swing client?

#17 - 03/20/2017 03:21 PM - Greg Shah

No matter if the Linux user name is the same as service owner name.

The Linux user name doesn't matter in that configuration, because there is no spawning occurring. Also, the client is running on a completely separate system so there is no use of win_spawn.exe at all. The FWD server is just being used via TCP/IP so it should work.

What this means is that the only problem with running as a windows service is with spawning. That means it only affects local spawning of the web client and any appserver agents or batch processes which are spawned. Since appserver agents and batch processes can be started manually, there is a workaround for them. The local web client is the only case that has no workaround.

Do I need to check another Windows machine as client to connect Windows server with Swing client?

No. It will work too, but we don't need to test it.

#18 - 03/20/2017 03:31 PM - Eugenie Lyzenko

Greg Shah wrote:

No matter if the Linux user name is the same as service owner name.

The Linux user name doesn't matter in that configuration, because there is no spawning occurring. Also, the client is running on a completely separate system so there is no use of win_spawn.exe at all. The FWD server is just being used via TCP/IP so it should work.

What this means is that the only problem with running as a windows service is with spawning. That means it only affects local spawning of the web client and any appserver agents or batch processes which are spawned. Since appserver agents and batch processes can be started manually, there is a workaround for them. The local web client is the only case that has no workaround.

Do I need to check another Windows machine as client to connect Windows server with Swing client?

No. It will work too, but we don't need to test it.

OK.