Map a network drive to be used by a service

Suppose some Windows service uses code that wants mapped network drives and no UNC paths. How can I make the drive mapping available to the service's session when the service is started? Logging in as the service user and creating a persistent mapping will not establish the mapping in the context of the actual service.

windows windows-services unc system-administration mapped-drive

edited Jan 10 '13 at 13:55
bluish
7,236 11 55 104

asked Oct 8 '08 at 13:29
VoidPointer
8,159 9 36 56

5 See ForcePush's answer, his workaround works. – Ubikuity Aug 26 '11 at 11:44

9 Answers

You'll either need to modify the service, or wrap it inside a helper process: apart from session/drive access issues, persistent drive mappings are only restored on an interactive logon, which services typically don't perform.

The helper process approach can be pretty simple: just create a new service that maps the drive and starts the 'real' service. The only things that are not entirely trivial about this are:

- The helper service will need to pass on all appropriate SCM commands (start/stop, etc.) to the real service. If the real service accepts custom SCM commands, remember to pass those on as well (I don't expect a service that considers UNC paths exotic to use such commands, though...)
- Things may get a bit tricky credential-wise. If the real service runs under a normal user account, you can run the helper service under that account as well, and all should be OK as long as the account has appropriate access to the network share. If the real service will only work when run as LOCALSYSTEM or somesuch, things get more interesting, as it either won't be able to 'see' the network drive at all, or require some credential juggling to get things to work.

answered Oct 8 '08 at 13:48
mdb
26.7k 6 47 59

1 Very informative... Am I correct in assuming that logon scripts are also run only for interactive logon sessions and not for service sessions? – VoidPointer Oct 8 '08 at 13:55

Use this at your own risk. (I have tested it on XP and Server 2008 x64 R2)

For this hack you will need SysinternalsSuite by Mark Russinovich:

Step one: Open an elevated cmd.exe prompt (Run as administrator)

Step two: Elevate again to root using PSExec.exe: Navigate to the folder containing

SysinternalsSuite and execute the following command `psexec -i -s cmd.exe` you are now inside of a prompt that is `nt authority\system` and you can prove this by typing `whoami`. The `-i` is needed because drive mappings need to interact with the user

Step Three: Create the persistent mapped drive as the SYSTEM account with the following command `net use z: \\servername\sharedfolder /persistent:yes`

It's that easy!

WARNING: You can only remove this mapping the same way you created it, from the SYSTEM account. If you need to remove it, follow steps 1 and 2 but change the command on step 3 to `net use z: /delete`.

NOTE: The newly created mapped drive will now appear for ALL users of this system but they will see it displayed as "Disconnected Network Drive (Z:)". Do not let the name fool you. It may claim to be disconnected but it will work for everyone. That's how you can tell this hack is not supported by M$.

edited Jan 10 '13 at 14:00
bluish
7,236 11 55 104

answered Jan 21 '11 at 20:11
ForcePush
1,379 2 6 2

2 I was trying to use this solution and ran into this problem: the mapped drive appears as disconnected to the users (even admins). Any suggestions? – Constantin Baciu Feb 14 '11 at 16:04

1 Works like a charm! Thanks! Been struggling with this for a while – Tommy Aug 4 '11 at 13:42

4 After a reboot the mapped drive is gone. Any ideas?The mapping is persisted, but the status is "Unavailable" so it doesn't show up – Tommy Oct 17 '11 at 9:48

2 I also see the mapping as unavailable after a reboot. – Dave Patteson Oct 20 '11 at 11:19

21 To get it working after a reboot, create a script just containing `net use z: \\servername\sharedfolder` and set it to run on computer startup, per technet.microsoft.com/en-us/library/cc770556.aspx This will run as the SYSTEM account, so no need for psexec. – TRS-80 Apr 20 '12 at 2:06

I found a solution that is similar to the one with psexec but works **without additional tools and survives a reboot**.

Just add a sheduled task, insert "system" in the "run as" field and point the task to a batch file with the simple command

```
net use z: \servername\sharedfolder /persistent:yes
```

Then select "run at system startup" (or similar, I do not have an English version) and you are done.

edited Jan 10 '13 at 14:02
bluish
7,236 11 55 104

answered Oct 23 '11 at 15:28
larry
309 3 2

what starts first? a windows service or this scheduled task? our service dies at start if the path is unavailable. We'd have to rebuild this if the service starts last. – Thomas Dec 20 '11 at 14:26

1 I tried this on 2008 R2, it creates a Disconnected Mapped Drive and when I try to open the same it says "z:\ is not accessible. Logon Failure:unknown user name or bad password." But I was able to create mapped drive successfully by running the same bat file manually. Ny insights? – Tech Jerk Jul 18 '13 at 7:32

@Thomas it doesn't matter which starts first providing the task has been run at least once because of the `/persistent:yes` – Edd Sep 25 '13 at 10:41

This the only method that worked for me on Windows 8.1. I had to do it to back up my NAS to Crashplan, and the trigger was set to happen when any user logged in. – Jim Hunziker Oct 27 '13 at 1:19

2 Why does it need to be a scheduled task? It is /persistent:yes, isn't that enough to keep it around? – Scott Stafford Feb 5 at 18:49

A better way would be to use a symbolic link using mklink.exe. You can just create a link in the file system that any app can use. See http://en.wikipedia.org/wiki/NTFS_symbolic_link.

answered Oct 26 '11 at 15:16


Hal
221 2 2

So simple and works well. Since it is part of the filesystem, the link is available to all accounts. Just make sure the service is running as an account that has access to the network resource (which might not be the case for System etc.). – Jeremy Frank Feb 23 '12 at 21:51

2 Too bad it does not work on 2000/XP. – To1ne Jan 16 '14 at 8:28

1 This is DEFINITELY the better answer to the question, thank you much! – Sten Petrov Jun 6 '14 at 14:32

You could us the 'net use' command:

```
var p = System.Diagnostics.Process.Start("net.exe", "use K: \\\\Server\\path");
var isCompleted = p.WaitForExit(5000);
```

If that does not work in a service, try the Winapi and PInvoke WNetAddConnection2

Edit: Obviously I misunderstood you - you can not change the sourcecode of the service, right? In that case I would follow the suggestion by mdb, but with a little twist: Create your own service (lets call it mapping service) that maps the drive and add this mapping service to the dependencies for the first (the actual working) service. That way the working service will not start before the mapping service has started (and mapped the drive).

edited Feb 5 at 19:06


Scott Stafford
17.4k 13 67 108

answered Oct 8 '08 at 13:38


Treb
12.3k 3 36 70

with this mapping service setup, I think that the drive mapping established by the mapping service wouldn't be available in the context of the original service, would it? – VoidPointer Oct 10 '08 at 8:51

I think it <disclaim>should</disclaim> - there is only one environment for each winlogon session. – Treb Oct 10 '08 at 15:45

You wan't to either change the user that the Service runs under from "System" or find a sneaky way to run your mapping as System.

The funny thing is that this is possible by using the "at" command, simply schedule your drive mapping one minute into the future and it will be run under the System account making the drive visible to your service.

answered Oct 8 '08 at 13:37


Torbjörn Gyllebring
9,718 2 20 22

the service does not run as "System". It is set up to run under a specific local account. Even I log in with that account, create a persistent network mapping, log out and restart the service, the mapping will not be available to the service. – VoidPointer Oct 8 '08 at 13:47

ForcePush,

NOTE: The newly created mapped drive will now appear for ALL users of this system but they will see it displayed as "Disconnected Network Drive (Z:)". Do not let the name fool you. It may claim to be disconnected but it will work for everyone. That's how you can tell this hack is not supported by M$...

It all depends on the share permissions. If you have Everyone in the share permissions, this mapped drive will be accessible by other users. But if you have only some particular user whose credentials you used in your batch script and this batch script was added to the Startup scripts, only System account will have access to that share not even Administrator. So if you use, for example, a scheduled ntbackuo job, System account must be used in 'Run as'. If your service's 'Log on as: Local System account' it should work.

What I did, I didn't map any drive letter in my startup script, just used `net use \\\server\share ...` and used UNC path in my scheduled jobs. Added a logon script (or just add a batch file to the startup folder) with the mapping to the same share with some drive letter: `net use Z: \\\...` with the same credentials. Now the logged user can see and access that mapped drive. There are 2 connections to the same share. In this case the user doesn't see that annoying

"Disconnected network drive ...". But if you really need access to that share by the drive letter not just UNC, map that share with the different drive letters, e.g. Y for System and Z for users.

edited Jan 23 '14 at 7:42

bluish
7,236 11 55 104

answered Jan 22 '14 at 15:14

lk7777
38 7

The reason why you are able to access the drive in when you normally run the executable from command prompt is that when u are executing it as normal exe you are running that application in the User account from which you have logged on . And that user has the privileges to access the network. But , when you install the executable as a service , by default if you see in the task manage it runs under 'SYSTEM' account . And you might be knowing that the 'SYSTEM' doesn't have rights to access network resources.

There can be two solutions to this problem.

1. To map the drive as persistent as already pointed above.
2. There is one more approach that can be followed. If you open the service manager by typing in the 'services.msc'you can go to your service and in the properties of your service there is a logOn tab where you can specify the account as any other account than 'System' you can either start service from your own logged on user account or through 'Network Service'. When you do this .. the service can access any network component and drive even if they are not persistent also. To achieve this programmatically you can look into 'CreateService' function at http://msdn.microsoft.com/en-us/library/ms682450(v=vs.85).aspx and can set the parameter 'lpServiceStartName ' to 'NT AUTHORITY\NetworkService'. This will start your service under 'Network Service' account and then you are done.
3. You can also try by making the service as interactive by specifying SERVICE_INTERACTIVE_PROCESS in the servicetype parameter flag of your CreateService() function but this will be limited only till XP as Vista and 7 donot support this feature.

Hope the solutions help you.. Let me know if this worked for you .

edited Dec 11 '12 at 5:03

Andro Selva
28.9k 24 119 176

answered Sep 29 '10 at 11:51

Kushagra
1

I found a really cool way to get UNC with credentials working in a windows service on codeproject.

see Adrian Hayes post: http://www.codeproject.com/Articles/43091/Connect-to-a-UNC-Path-with-Credentials

His solution works a treat.

answered Dec 3 '13 at 14:34

John
578 6 19