xCmd an Alternative to PsExec

Published: 2011-04-12 · Archived: 2026-04-05 19:15:41 UTC

I recently had to use PsExec for one of my project and it had to integrated into program. I mean invocation of PsExec had to be done from a coding language. Then i faced problem of application getting hanged. To verify my code i did call PsExec from Java, Csharp, VB.net. None of them worked and program hanged on calling PSExec.

There is a problem with PsExec OutputStream when it is called from a programming language. None of the versions of PsExec looked good with this bug. So i had tpo find out an alternative, Here is one good tool named XCmd by "Zoltan Csizmadia"

Execute Applications on Remote Systems

- Overview

This program allows you to execute applications on remote systems without installing any client software. You can start a command prompt or just execute a command or exe on a remote machine. The only restriction is you must be an administrator 😦

Everybody knows the cool tools from Sysinternals (www.sysinternals.com). One of my favorites are PSEXEC, PSKILL and PSLIST,… 🙂
I was always wonder how they could query every kind of information or execute commands on a remote machine without installing any client software.

Features

- With this program you can run as many remote commands on the remote machine as you want. (PSEXEC supports only one remote command on the remote machine at the same time)
- You can execute internal commands (dir,..) directly.
  xCmd.exe \\remote dir
- You can start a light "telnet" connection with a remote machine without any telnet server
  xCmd.exe \\remote cmd

Usage

xCmd v1.0 for NT4/2000 – executes commands remotely
Freeware! 2001 Zoltan Csizmadia, zoltan_csizmadia@yahoo.com

Usage: xCmd.exe \\computer [options] command/exe arguments

Options:
/D:directory Set working directory
Default: Remote "%SystemRoot%\System32"

/IDLE Idle priority class
/NORMAL Normal priority class
/HIGH High priority class
/REALTIME Realtime priority class
/C Copy the specified program to the remote machine's
"%SystemRoot%\System32" directory
Commands's exe file must be absolute to local machine
/USER:user User for remote connection
/PWD:{password|*} Password for remote connection
/NOWAIT Don't wait for remote process to terminate

Examples:
xCmd.exe \\remote cmd
xCmd.exe \\remote /user:administrator dir c:\
xCmd.exe \\remote /user:somebody /pwd:* /d:d:\ test1.exe /p1 /p2
xCmd.exe \\remote /c /user:somebody /pwd:* /d:d:\ test2.exe /whatever

– Input is passed to remote machine when you press the ENTER.
– Ctrl-C terminates the remote process
– Command and file path arguments have to be absolute to remote machine
If you are using /c option, command exe file path must be absolute to
local machine, but the arguments must be absolute to remote machine

How does it work?

1. The xCmd.exe is console application and when you start it, the program will extract a xCmdSvc.exe from its resources.
2. xCmd.exe creates a service on the remote machine (that's the reason, you must be an administrator
3. xCmd.exe starts the remote service (#2)
4. xCmd.exe and xCmdSvc.exe will communicate via named pipes
5. xCmd.exe send a packet to the service what to execute
6. xCmdSvc.exe starts the command and redirect stdout, stderr, stdin to 3 named pipes.
7. xCmd.exe listens these 3 named pipes (#6), redirect them to its stdout, stderr, stdin

I have downloaded his code and converted to Visual Studio 2008 Solution and fixed a small bug which comes with iostream.h inclusion in new version of C++.

Here is the link to original Article which i copied

http://www.codeguru.com/Cpp/I-N/network/remoteinvocation/article.php/c5433

And here is modified C++ Solution which gets compiled in Visual Studio. Checkout from svn

https://linkwithweb.googlecode.com/svn/trunk/Utilities/RemoteExecution/xCmd

---

Source: https://ashwinrayaprolu.wordpress.com/2011/04/12/xcmd-an-alternative-to-psexec/