What's the Windows equivalent of a UNIX shell script?

Asked 16 years, 1 month ago Modified 7 years, 11 months ago Viewed 68k times

28

I want to have an executable file that will call some other programs. The way I would do this in Linux is with a simple bash script that looks like this:

```
#!/bin/bash
echo "running some-program"
/home/murat/some-program arg1 arg2
```

What's the best way to do this kind of thing under Windows?

windows command-line shell batch-file

Share Improve this question Follow

asked Nov 17, 2008 at 21:03

Murat Ayfer

3,914 ● 6 ● 31 ● 26

15 Answers

Sorted by: Highest score (default)

31

✓

Take a look at PowerShell, which is the closest you will get to a true scripting language like you have in Unix. Other than that, for simple things such as simply runnning an application, take a look at Windows command script/MS-DOS batch files.

Share Improve this answer Follow

answered Nov 17, 2008 at 21:05

Scott Dorman

42.5k ● 12 ● 81 ● 112

The primary issue with PowerShell (for me, at least) is that it is currently not installed by default on most versions of Windows (including XP and Vista). However, it will be included out-of-the-box on Windows Server 2008 and on Windows 7: tinyurl.com/5jhh2f
– Jon Schneider Nov 18, 2008 at 18:26

1 Yes, PowerShell is not installed by default on any versions of Windows except Win7, which will actually be PowerShell v2. – Scott Dorman Nov 18, 2008 at 23:29

17

You could go with either an old school batch file (http://en.wikipedia.org/wiki/Batch_file) or you can go with a brand new shiny Windows PowerShell script (http://www.microsoft.com/technet/scriptcenter/hubs/msh.mspx).

I use batch for simple operations and PowerShell for complex scripts that also involve data processing.

Share Improve this answer Follow

answered Nov 17, 2008 at 21:07

Bogdan

3,075 ● 1 ● 23 ● 21

12

There are a number of scripting options on windows:

- Batch files
- VBScript/CScript/JScript (very powerful when combined with WMI)
- PowerShell
- You could install Perl, Python, or other engine
- You can install unix tools, via Cygwin for example, to get make most of the common unix commands available in windows as well.

- Since we're all (mostly?) programmers here, there's nothing stopping you from making a .exe file to do what you want.

Share
Improve this answer
Follow

edited Nov 17, 2008 at 23:56

answered Nov 17, 2008 at 21:15
Joel Coehoorn
415k ● 114 ● 577 ● 813

1 Just to be clear, using powershell requires an install right now as well. – EBGreen Nov 17, 2008 at 21:17

10

You can use a Windows batch file. Find more information in the Wikipedia article.

Share
Improve this answer
Follow

edited May 11, 2013 at 6:57
Cody Gray ♦
244k ● 52 ● 501 ● 581

answered Nov 17, 2008 at 21:05
Eric Wendelin
44.3k ● 9 ● 72 ● 94

7

The Windows NT family of Operating Systems (NT 3.51, NT 4.0, 2000, XP, Vista) have a reasonably powerful Command Shell even if it is a bit esoteric. This shell does not have the power of the typical UNIX shells however it is ubiquitous on Windows machines and it is pretty amazing what can be achieved. A great book for this is Windows NT Shell Scripting by Timothy Hill. I always found Rob van der Woude's Scripting Pages a great resource for batch files.

PowerShell is Microsoft's latest offering in this field and takes scripting to a whole new level and gives you access to all of the power of the .NET framework as well as WMI, COM and any other interface into Windows. There is a pretty steep learning curve with this but it is definitely the way to go and I think it has the potential to become one of those foundational technology skills (like RegEx, XPath etc) in the Microsoft world. A good starting point is Microsoft Windows PowerShell Technology Center but there are a huge number of resources out there for this.

Share
Improve this answer
Follow

edited Apr 17, 2009 at 6:12

answered Nov 17, 2008 at 21:23
Martin Hollingsworth
7,299 ● 7 ● 51 ● 52

1 Rob vandr Woude's Scripting pages are now here: robvanderwoude.com/batchfiles.php – JohnnyLambada Apr 16, 2009 at 21:28

6

I used to use batch files. And for simple examples like you gave above I still do, but more and more I have been using python to do these sorts of things. It has the added benefit of working across platforms for those rare scripts that are actually meaningful on multiple platforms.

Python is just one of many scripting languages you could use for this sort of thing. Perl, ruby, or php may be more your speed. I find that a scripting language gives me a richer and easier to understand syntax if I want to do something a little more complicated than just start a program with a few default arguments.

Share Improve this answer Follow

answered Nov 17, 2008 at 21:09

grieve

13.5k ● 10 ● 52 ● 62

6

As long as we're suggesting *nix command-line tools, I'd recommend checking out Cygwin (http://www.cygwin.com) as it emulates a *nix-like environment.

Share Improve this answer Follow

answered Nov 17, 2008 at 21:14

Eric Wendelin

44.3k ● 9 ● 72 ● 94

5

Update: Wow, this is woefully out of date now (though that's inevitable). Powershell is definitely becoming more and more the language of choice for automation in Windows. It has many of the advantages of any .net language (like C#) combined with much of the expediency that exists in any good shell language, without the serious downsides of batch files.

For what it's worth, the most popular options I've seen at a certain large, Redmond based software company (in rough order of popularity):

- Batch files
- Perl
- Windows Scripting Host (Javascript)
- Tiny exes

Much less popular:

- Powershell

Note that Perl doesn't require an install, it only requires a few environment variable updates, so it's easy to just shove a full distro into your source control system along with other tools.

My personal preference would be to avoid batch files as much as possible, even with the most careful and rigorous of processes the benefit for the cost is just not worth it. You'll never end up with a satisfactory solution if you use batch. Perl and Javascript are tempting to use but there are troublesome fundamental deficiencies which leave a lot to be desired, without strong development guidelines these methods will more likely than not lead to low quality code.

Python and Powershell are good candidates but I'm not sufficiently adept with them and, more importantly, a lot of people aren't that familiar with them. Perhaps that's not a good enough excuse. Tiny exes can have many problems as well and require discipline and process in order to ensure that the source is always checked in, that the exes conform to certain UI norms (e.g. they have useful help if you run -? or /?), that they have an obvious owner, that the process for updating the exes is discoverable, etc.

Share

Improve this answer

Follow

edited Mar 16, 2013 at 6:47

answered Nov 17, 2008 at 23:25

Wedge

19.8k ● 7 ● 50 ● 71

4

batch files work for me. Just remove the first line in your example and it should work the same in win32 cmd line.

note - make a text file, name it with a .bat extension and use a backslash rather than a forward slash.

Share

Improve this answer

Follow

edited Nov 17, 2008 at 21:12

answered Nov 17, 2008 at 21:05

Tim

20.4k ● 24 ● 122 ● 219

1 He'll need to fix the slashes – sblundy Nov 17, 2008 at 21:07

4

As an alternative to the underpowered Windows command line, the not very Windows-friendly Cygwin or the quirky and slow Powershell, you might consider [Hamilton C shell](). I am the author, so of course I'm possibly biased, but I am a real person and I do care about trying to make my customers happy. Hamilton C shell is a complete recreation of the UNIX C shell and utilities but all of it's from scratch for Windows and, as such, it's designed to follow the Windows conventions for filenames and so on. I just released a new version this summer for Vista that includes both 32-

bit and 64-bit builds. (The 64-bit version is about 2x to 3x as fast as the 32-bit version.) There's a free demo, so you can try before you buy.

Share Improve this answer Follow

answered Sep 22, 2009 at 16:36

Nicole Hamilton

51 ● 2

2

While shell have `sh -C ${custom command}`.

Try on `cmd /C ${custom command}` where ***${custom command}*** stand for the command you want to get executed.

Share Improve this answer Follow

answered Dec 23, 2016 at 14:54

Salathiel Genese

1,909 ● 2 ● 23 ● 38

0

Generally, the windows command line (DOS) is considered rather impoverished compared to *nix. You could install something like Cygwin or PowerShell if you want a powerful command shell, or you could cause yourself great pain and suffering to try using DOS for all your needs. :)

Share Improve this answer Follow

answered Nov 17, 2008 at 21:09

Ian Varley

9,437 ● 5 ● 31 ● 34

1 Every Windows command script I've written in recent years won't run on DOS. The only thing you're inheriting from DOS are parts of the syntax and commands. – Joey Apr 17, 2009 at 11:26

0

WScript.CreateObject("Scripting.FileSystemObject");

has been my win32 weapon of choice as of late.

Share Improve this answer Follow

answered Nov 17, 2008 at 21:17

Nick Van Brunt

15.5k ● 13 ● 71 ● 92

I use a combination of CScript/JScript. Very powerful once you learn the available system objects.

0

Share Improve this answer Follow

answered Nov 17, 2008 at 21:19

Rob
78.5k ● 57 ● 161 ● 199

0

Here is an example batch file I made for someone a while back.

```
echo off
echo Backing up Images
%homedrive%\Backup\7z.exe a -tzip photos.zip %userprofile%\Photos\*.jpg
%userprofile%\Photos\*.png %userprofile%\Photos\*.gif
%userprofile%\Photos\*.bmp
echo Backing up Music
%userprofile%\Backup\7z.exe a -tzip music.zip %userprofile%\Music\*.mp3
%userprofile%\Music\*.wma %userprofile%\Music\*.ogg %userprofile%\Music\*.wav
echo All Done
```

Note that the script expects to have the 7z.exe program in a folder called Backup in the %homedrive% location. I used Envirnment Variables to make it easier for different users.

Share Improve this answer Follow

answered May 11, 2013 at 7:35

Ben P. Dorsi-Todaro
321 ● 1 ● 6 ● 20