

How to Use Windows Batch File Commands to Automate Repetitive Tasks

By Gavin Phillips

Read the original article here: <http://www.makeuseof.com/tag/use-windows-batch-file-commands-automate-repetitive-tasks/>



Before Windows became **our favorite GUI**, everything was done using commands. Some of our readers may remember using MS-DOS commands to complete the smallest of tasks. These days, you can still use commands to automate tasks and **speed up your productivity**.

If you have a number of boring and repetitive tasks, a batch file **is exactly what you're looking for**. Instead of manually executing tasks, a single batch file can be written to take care of everything.

The Basics: Creating a Batch File

Batch files are written in plain text. You can use whichever plain text editor you like, but the standardized Notepad does the job just fine. If you're creating really complex programs, it can be handy to have the **additional features of Notepad++**, but for our examples, we will stick with what we know.

Each example batch file provided has been **tested using Notepad**. Once we've finished inputting our commands, we'll head to **Save As**, and give the file an appropriate name. Once saved, we can change the file extension from **.txt** to **.bat**, and press **Enter**. You should notice the file icon immediately change to an application window featuring a cog.

Tasks You Can Automate

Here are a few really useful batch commands for you to play around with, and some short descriptions of what **each command syntax and parameter can do**.

1. Open Multiple Programs

This one will let you start multiple programs with a single click. First, identify the programs or files you want to open simultaneously. In this instance, I'll be opening Chrome, **a Word file** I'm working on, and VMware Player.

Open a new text file and type:

```
@echo off

cd "C:\Program Files\Google\Chrome\Application\"

start chrome.exe

start - "C:\Program Files\Microsoft Office\Office15\WINWORD.EXE"

"C:\Work\MUO\How to Batch Rename.docx"

cd "C:\Program Files (x86)\VMware\VMware Player"

start vmplayer.exe

exit
```

Of course, you can add numerous applications and files to this list. For super-opening powers, you can **pair this batch file with a hotkey**.

```

StartPrograms.bat - Notepad
File Edit Format View Help
@echo off
cd "C:\Program Files\Google\Chrome\Application\"
    start chrome.exe

start - "C:\Program Files\Microsoft Office\Office15\WINWORD.EXE"
    "C:\work\MUO\How to Batch Rename.docx"

cd "C:\Program Files (x86)\VMware\VMware Player"
    start vmplayer.exe
exit

```

The commands we have used here are:

- **@echo** displays the command currently being executed in a command shell. We turned this **off**.
- **cd** changes the directory
- **start** does the obvious, and starts the program

2. Delete Files Older Than

We can also use a batch file to delete files older than a user-set amount of days. This can be used to delete one specific file type, or a group of files in a folder, so long as they meet the specifications in the batch file. Our first example will delete files in the specified folder older than three days:

```
forfiles /p "C:\some\file\name\here" /s /m * /d -3 /c "cmd /c del @path"
```

Our second example will delete only files with the extension .docx, again older than three days:

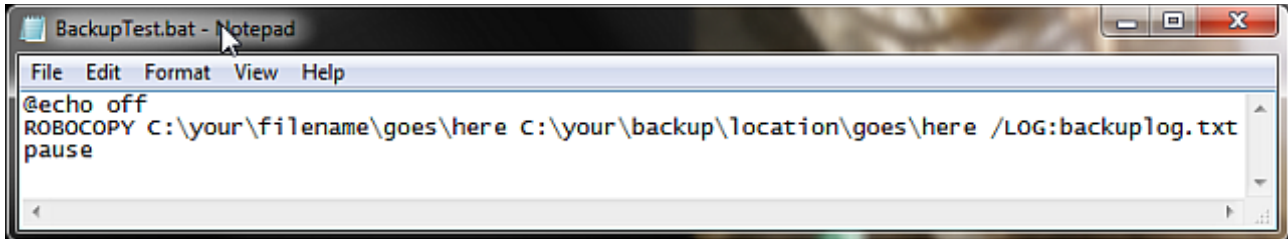
```
forfiles /p "C:\some\file\name\here" /s /m * .docx /d -3 /c "cmd /c del @path"
```

The commands and switches we have used here are:

- **forfiles** allows us to use commands for each file in a location i.e. the commands will apply to each file fitting the command arguments
- **/p** details the path to start searching i.e. the directory you want to delete the files from
- **/s** instructs the command to search sub-directories
- **/m** instructs the command to use the given search mask. We used the wildcard operator "*" in our first example, and specified **.docx** in the second
- **/d-3** is the time setting. Increase or decrease depending on your requirements
- **/c del @path** is the delete aspect of the command

3. Backing Up Your System

We can use batch files to back up specific folders, or as part of **a wider backup operation**. Most people have backups and system restore points setup as standard, but sometimes it pays to make a couple of copies of anything that might make you cry if it were deleted. I'm going to show you one very simple method and another slightly more advanced version.



```
BackupTest.bat - Notepad
File Edit Format View Help
@echo off
ROBOCOPY C:\your\filename\goes\here C:\your\backup\location\goes\here /LOG:backuplog.txt
pause
```

Method #1

Open Notepad. Type the following, following the instructions:

```
@echo off

ROBOCOPY C:\your\filename\goes\here C:\your\backup\location\goes\here /
LOG:backuplog.txt

pause
```

Save the file, then rename to systembackup.bat, and press **Enter**.

This method works well when backing up individual folders, but isn't entirely practical for anything more complex. The commands used here are:

- **ROBOCOPY** stands for Robust File Copy, which replaced when Windows Vista was released

Method #2

This time we will build a longer string of folders we want to back up, including your Outlook address book, and your system registry.

```
@echo off

:: variables

set drive=X:\Backup

set backupcmd=xcopy /s /c /d /e /h /i /r /y

echo ### Backing up My Documents...

%backupcmd% "%USERPROFILE%\My Documents" "%drive%\My Documents"

echo ### Backing up Favorites...

%backupcmd% "%USERPROFILE%\Favorites" "%drive%\Favorites"

echo ### Backing up email and address book (Outlook Express)...
```

```
%backupcmd% "%USERPROFILE%\Application Data\Microsoft\Address Book" "%drive%
\Address Book"

%backupcmd% "%USERPROFILE%\Local Settings\Application Data\Identities" "%drive%
\Outlook Express"

echo ### Backing up email and contacts (MS Outlook)...

%backupcmd% "%USERPROFILE%\Local Settings\Application Data\Microsoft\Outlook"
"%drive%\Outlook"

echo ### Backing up the Registry...

if not exist "%drive%\Registry" mkdir "%drive%\Registry"

if exist "%drive%\Registry\regbackup.reg" del "%drive%\Registry\regbackup.reg"

regedit /e "%drive%\Registry\regbackup.reg"

echo Backup Complete!

@pause
```

A little explanation to what's going on here: First, we set the location we want the files to be copied to – “**set drive=X:\Backup**” – followed by the backup command we are going to use – **xcopy**. Following the xcopy command is a string of parameters telling the batch file to perform a number of tasks:

- **/s** copies system files
- **/c** carries out the command specified by the string, then terminates
- **/d** enables drive and directory changes
- **/e** copies empty directories
- **/h** copies hidden files
- **/i** if destination doesn't exist, and you're copying more than one file, **/i** assumes the destination must be a directory
- **/r** overwrites read-only files
- **/y** suppresses prompts confirming you want to overwrite read only files

To add more folders to the batch file, use the following syntax:

```
%backupcmd% "...source directory..." "%drive%\...destination dir..."
```

We've selected a number of folders to copy, but you might note they are all part of your USERPROFILE. If you just want to backup the entire folder, you can use this command, assuming the same “set drive” and “set backupcmd” are in use:

```
%backupcmd% "%USERPROFILE%" "%drive%\%UserName% - profile"
```

4. Change Your IP Address

Most of the time **your laptop will use a dynamic IP** address to connect to the Internet. Sometimes, you might be **required to use a static IP**, for instance at your workplace, school, or other “official” locations. Sure, you could manually change it, but if it’s somewhere you visit regularly, why not make a batch file to do the work for you?

For this we’ll create two files: one to make it static, and another to change it back to dynamic.

Static

You should type the following for your static batch file:

```
netsh interface ip set address "LAN" static "xxx.xxx.xxx.xxx" "xxx.xxx.xxx.x"
"xxx.xxx.xxx.x"
```

Where the first series of “x’s” is your required static IP, the second is the network/subnet mask, and the third is your default gateway.

Dynamic

This batch file will return your network adaptor settings to a dynamic setup:

```
netsh int ip set address name = "LAN" source = dhcp
```

While neither of these files are doing anything by batch, just having those files can be a timesaver. If you have more than one network to connect to, copy the first file and alter the details.

5. Make Your Kids Go to Bed

My kids aren’t old enough to be playing video games in the middle of the night, but I remember my own tactics against my parents so **I could play Championship Manager 2** into the small hours of the morning. Luckily, my parents didn’t know about using commands to control my actions. Try this:

```
@echo off

:W

If %time%==23:30:00.00 goto :X

:X

shutdown.exe /s /f/ t/ 120 /c "GO TO BED RIGHT NOW!!!"
```

Here, the computer constantly checks to see if the time is half-past eleven. When the time correlates, the message “GO TO BED RIGHT NOW!!!” will display, along with the 120s countdown timer. The 120s *should* be enough time to save whatever game they are playing, or their work, before the computer shuts down.

To stop this countdown, press **Win+R**.

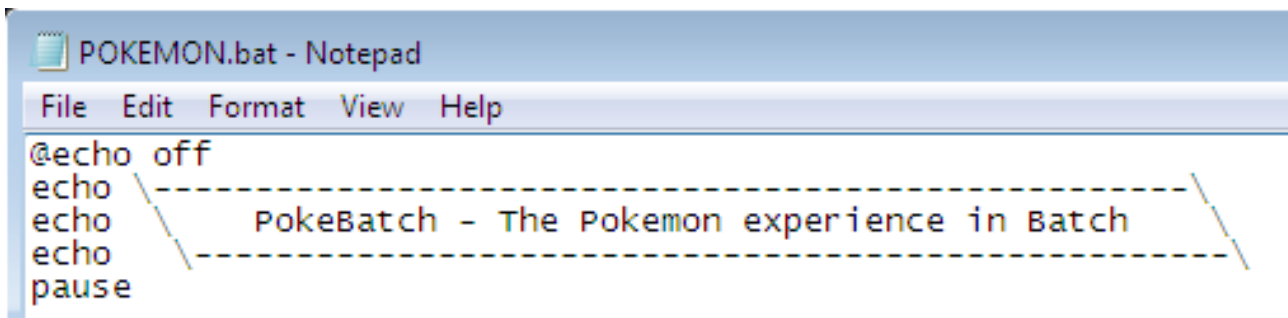
Of course, don’t tell the children this.

6. Batch Rename & Mass Delete

I've written a more extensive article dealing with [batch file renaming and deletion](#), so I won't explore this one too much, but batch files can be used to automate these sometimes tedious tasks. Check out the article for some extended batch commands, and get bulk deleting straight away.

Bonus: Pokémon!

So this has [nothing to do with productivity](#), and if you're susceptible to [Pokémon-related gaming addictions](#), maybe don't download and play this. If not, you can grab this text-based adventure [right here](#).



```
@echo off
echo
echo
echo
echo
pause
```

PokeBatch - The Pokemon experience in Batch



Whetted Your Appetite?

These are just six batch files you can create to automate tasks on your system. If you [learn a little more](#), you'll be able to accomplish unheralded amounts of activities on your system between batch files and the Command Prompt. And if that doesn't suit you, you could always take a [look at the Windows PowerShell](#).

What are your favorite batch files? What do you automate to shave precious minutes from your day? Let us know [in the comments!](#)

Read more stories like this at

