

Working at a command line prompt

Working at a Command Line Prompt is a skill which is needed by all working technicians. It is also a requirement for A+ *Essentials*, A+ *Technician* (602) and A+ *Remote Support Technician* (603) exams. The A+ *Essentials for Students* and A+ *Electives for Students* books introduce command line work in sufficient detail for the exams. The purpose of this tutorial is to go beyond the minimum requirements for the exams and to look at command line work in a little more detail.

The modern Windows versions make it possible for you to do just about anything by pointing and clicking with a mouse, so for most users – even those who might describe themselves as 'power users' – the command line prompt is unnecessary. Until, that is, the Graphical User Interface (GUI) fails or you want to write a script to automate an incremental backup, or re-set the attributes of several hundred files in a matter of seconds, or use a Recovery Console.

CMD.EXE and COMMAND.COM

Both Windows 2000 and XP use a command line processor called CMD.EXE by default; they also support the use of the older COMMAND.COM as a legacy from the days of Windows 9.x and the original DOS. This tutorial will concentrate on the modern CMD.EXE command line processor. Examples have all been done using Windows XP Professional: other Windows versions are all but identical in operation.

1. Getting started

Both Windows 2000 and XP implement the Command line prompt as one of the *System Tools* and if you wish to launch a prompt, navigate through the **Start | (All) Programs** menu and click on the **Command Prompt** entry. You may want to make a

Desktop or Start Menu short cut to this to make access quicker. It is also possible to change the default (white on black) appearance of the prompt. To do this:

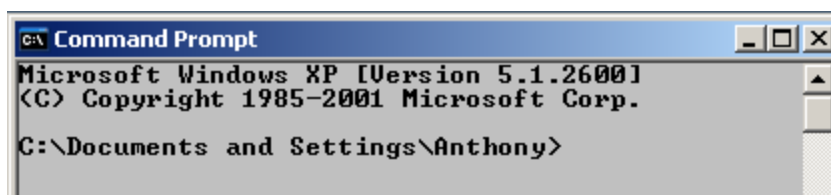
1. Right click on the menu entry or short cut.
2. Select **Properties** from the context menu.
3. Select the **Colors** tab.
4. Select the colour combination that you want.
5. Click on the **Apply** or the **Okay** button.

Next time you click on the short cut the Command line prompt will appear with your choice of colours implemented.

Paths

You will probably recall from your A+ studies that the file system on a Windows PC consists of a number of Directories (Folders) and Subdirectories which are relative to the 'root' of a Windows disk 'drive'. A drive is simply a disk partition with a file system and a drive letter which identifies it. Every Windows PC that has a working hard drive will have as its first hard drive a partition which is identified as C:. When you launch a Command prompt, it indicates your current position in the file system hierarchy and the drive to which you are attached. Figure 1 shows the Command line prompt on a Windows XP machine. You can close the prompt window by clicking on the square button at the top right of the windows or by typing the command EXIT [+Carriage Return] at the prompt.

Figure 1.



By default, the Command line processor will open in the 'Home' directory of the currently logged in user on the C: drive and will display the version of Windows that you are running. If you are happy with this, you need do nothing: if you want to change it you can do so by using some straightforward commands.

Navigating at a prompt

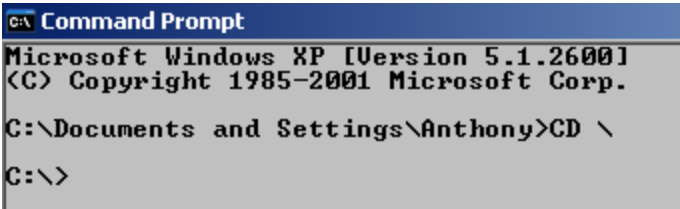
In order to move to a different directory use the Change Directory (CD) command.

The form of this command is CD (or CHDIR) followed by the name of the target directory. If the target directory exists the prompt will change immediately after the command has run. If we are currently logged on the the default directory on the C: drive (as in Figure 1) the one directory that we *know* exists is the root directory of that drive. The root directory of any drive is indicated by a single backslash character '\ so to change from the current directory – whatever it is – use the command

```
CD \ [+Carriage Return]
```

(Note the space between the command (CD) and its 'argument' (\) and the need for a Carriage Return to make the command run. Applying the 'CD \' command line to the example in Figure 1 will result in the output shown in figure 2.


Figure 2



```
C:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Anthony>CD \
C:\>
```

The prompt now indicates that we are 'in' the root directory of the C: drive.

If you use the DIR [+CARRIAGE RETURN] command at the C:\ prompt you will see a DIRectory listing of the root directory of C: as shown in figure 3.

Figure 3

```
C:\>DIR
Volume in drive C has no label.
Volume Serial Number is 008E-E804

Directory of C:\

18/11/2006  09:16                0  AUTOEXEC.BAT
19/11/2006  03:00            12,219,983  AUG7QT.DAT
18/11/2006  10:52        <DIR>          CmdScripts
18/11/2006  09:16                0  CONFIG.SYS
20/01/2007  00:52            87,287  debug.log
18/11/2006  09:22        <DIR>          Documents and Settings
13/12/2006  18:55        <DIR>          EPSON
01/02/2007  13:57        <DIR>          Program Files
09/02/2007  08:37        <DIR>          StudyMate Projects
13/12/2006  09:46            9,101  systemlog.txt
22/01/2007  12:16        <DIR>          TEMP
11/02/2007  17:43        <DIR>          WINDOWS
               5 File(s)      12,316,371 bytes
               7 Dir(s)  14,071,885,824 bytes free

C:\>
```

The directory listing shows the times, dates and sizes of the listed files and indicates subdirectories with the <DIR> marker.

The root directory of C: contains a lot of system files (some of them hidden and not listed by the DIR command) and it is generally not thought to be a good idea to use the root of the C: drive for storing user created files. The answer to this, of course, is to create a working directory of your own. To do this, use the MD (or MKDIR) command and give it the name of your new directory as its argument. To create a new directory MyWorkArea as a subdirectory of the root:

1. Make sure that you are in the root directory (use CD \ if you are not).
2. Issue the command MD MyWorkArea (Don't forget the Carriage Return).
3. Issue the command DIR to confirm that your new directory has been made.

Issue the command CD MyWorkArea.

The C: prompt will now appear as shown in figure 4.

Figure 4

```
C:\>
C:\MyWorkArea>_
```

Note that although the Windows command line is NOT case sensitive (DIR dir and DiR are all the same) it does preserve the case of what you enter, so when creating a directory you can use a mixture of upper and lower case letters for your name to make it easier to read. It is also possible to have spaces in names – My Work Area – would be acceptable, but spaces require you to enclose the name in quotes when using it as in: CD “My Work Area”; this is a complication that most techs would rather avoid.

Customising your CMD prompt

Having created your work area, you can save time when using the prompt in future by causing it to open in your work area when you click on the icon to launch it. To do this exit from the command line – type EXIT at the prompt – or just close the window, then:

1. Right click on the command prompt icon and select **Properties** from the context menu.
2. Click on the **Shortcut** tab. The Start in field will contain an entry which points to the default %HOMEDRIVE%%HOMEPATH%
3. Change this to your new directory C:\MyWorkArea
4. Click on the **Okay** button

Next time you click on the icon the CMD processor will be launched and you will be in your MyWorkArea directory.

Getting help

Both XP and 2000 provide on-line help functions for the CMD processor. To obtain a list of the main commands on the system use the HELP command at a prompt. To obtain more detailed help on a particular command use the command name followed by a front slash and a question mark or HELP [command name]. So, to get help on the DIR command for example type either HELP DIR or DIR /? at a prompt.

Summary (1)

In this first section we have seen:

- How to launch a CMD processor.
- How paths work in Windows.
- How to change the appearance of the command line window.
- How to use the DIR command to list the contents of a directory.
- The commands needed to change directory and to make a new directory
- How to change the default working directory where the CMD processor will start.
- How to get help on commands.

2. Working with directories and files

Directories

We have seen how to create directories and how to navigate between them. However, there are other commands that can be used to manage both directories and files. In order to remove a directory, use the RD (or RMDIR) command like this:

```
RD [target directory] [+Carriage Return]
```

This will remove the target directory *provided that it is empty*. In order to remove a directory and all its contents we need to modify the action of the RD command by using a command line *switch*. Switches are distinguished by prefixing them with a front slash. In order to see how this works, create a directory structure in your work area and then remove it with the command RD with the /S (for subdirectory) switch.

1. Launch a CMD processor in your **MyWorkArea** directory.
2. Make a subdirectory **MySubOne** (MD MySubOne).
3. Change directory to **MySubOne** (CD MySubOne).
4. Make another subdirectory **MySubTwo** (MD MySubTwo).

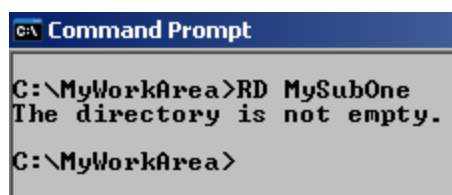
You now have a directory 'tree' which can be written in full as:

C:\MyWorkArea\MySubOne\MySubTwo

To test the working of the RD command, try this:

1. Navigate to your MyWorkArea directory.
2. Issue the command RD MySubOne. You will see an error message like the one in figure 5.

Figure 5



```
C:\ Command Prompt
C:\MyWorkArea>RD MySubOne
The directory is not empty.
C:\MyWorkArea>
```

3. Now try again with the /S switch. RD /S MySubOne

You will be prompted to make sure that you really want to do this:

Figure 6



```
C:\ Command Prompt - RD /S MySubOne
C:\MyWorkArea>RD /S MySubOne
MySubOne, Are you sure (Y/N)? _
```

4. Answer 'N'
5. Issue the command RD /S /Q MySubOne. Because you have used the 'Q' (for quiet) switch the system simply removes the directory and all its contents – files and subdirectories – without warning. Use this option only if you know *exactly* what you want to do.

Files

As you will be aware from your A+ studies all files on a Windows system have

'attributes' and these can be listed and / or changed by using the ATTRIB command. If

you use the ATTRIB command without arguments, files can be seen to have the attributes:

- Read Only
- Archive
- System
- Hidden

All of these attributes can be clear or set. Figure 7 shows an ATTRIB listing of the root directory of the C: drive on a fairly typical XP PC.

Figure 7

```
C:\>ATTRIB
A          C:\AUTOEXEC.BAT
A          C:\AUG7QT.DAT
          SH C:\boot.ini
A          C:\CONFIG.SYS
A          C:\debug.log
A SHR     C:\IO.SYS
A H       C:\IPH.PH
A SHR     C:\MSDOS.SYS
A SHR     C:\NTDETECT.COM
A SHR     C:\ntldr
A SH      C:\pagefile.sys
A         C:\systemlog.txt
C:\>
```

Files with the **S**ystem, **H**idden or **R**ead Only attribute set are usually (as in this case) important system files that are probably best left alone; though if you *insist* on fiddling with them read the HELP on the ATTRIB command and take your chances! Other files, the sort that you may create yourself, will have the **A**rchive attribute set; this merely indicates that the file has not yet been backed up or has been modified since it was last backed up.

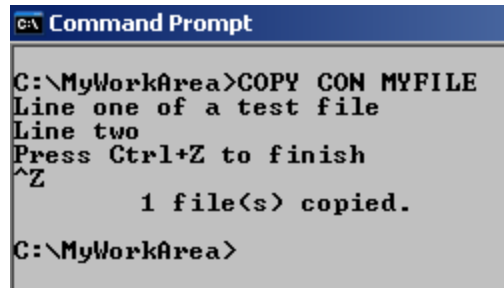
Making a test file (or two)

1. Start a CMD processor in your MyWorkArea directory.
2. Type COPY CON MYFILE [CR]
3. Type in some text.

4. When you have added some lines of text and you have finished, press Control+ Z.

The process looks like this:

Figure 8



```
C:\MyWorkArea>COPY CON MYFILE
Line one of a test file
Line two
Press Ctrl+Z to finish
^Z
        1 file(s) copied.
C:\MyWorkArea>
```

The DIR command will list your new file.

The ATTRIB command will show its attributes.

The TYPE command: TYPE MYFILE will display its contents.

5. Make as many files as you want using this method. Hint: as these are text files you could give them the .TXT extension which will identify them to the operating system as text files and will make it easy to open and edit them through the Windows GUI with the default Notepad editor if you wish.

DELEting a file

A file is deleted by using the DEL (for delete) command or its alias ERASE. Either way, the usage is DEL (or ERASE) FILENAME.EXT. Provided that the file is not Read only, System or Hidden, the system will delete it without asking *Are you sure?* To see this in action (so to speak) use the DEL command to delete the test file that you made earlier.

```
DEL MYFILE
```

Then use the DIR command to see that it has been removed.

Working with wildcards

It is often useful to be able to specify a group of files as the argument to a command such as DEL and for this, there are two special characters: the question mark ? and the asterisk or star *. The ? symbol stands for any individual character and the * for any group of characters. To specify *all* files with *all* extensions, simply use the designation *.* Thus, to delete all files in the current directory you would use the command DEL *.* or - to delete all files with the extension .TXT - DEL *.TXT. To delete all files in the current directory *and* in all subdirectories underneath it use DEL *.* /S Obviously, you need to be careful when using wildcards, particularly when specifying files to delete: there is no 'Recycle Bin' at a command prompt and what's done can't be undone!

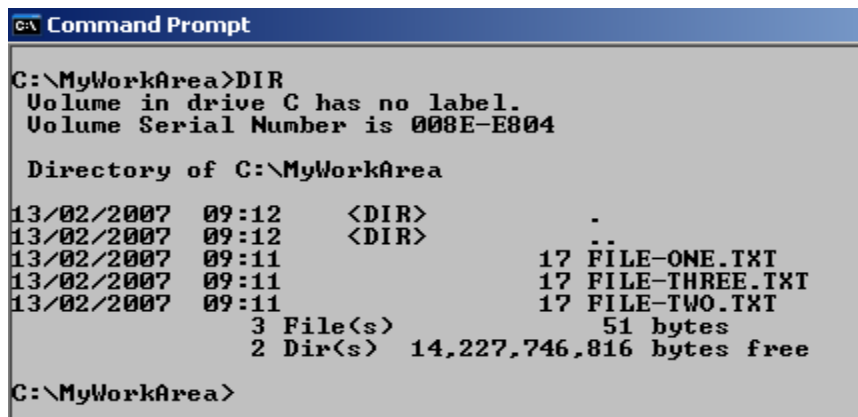
Practical exercise(s) with files

Any of the techniques described in this tutorial should be tried wherever and whenever possible. However, for this section we need to create some files so that we can work with them in a fairly systematic way. The exercise assumes that you have a PC with a working floppy drive A:. If this is not the case then you will have to modify things to suit your particular setup.

1. Create three text files in your working directory on C: drive. In this example these are called: FILE-ONE.TXT, FILE-TWO.TXT and FILE-THREE.TXT.

A DIRectory listing will look (something) like this:

Figure 9



```
C:\ Command Prompt
C:\MyWorkArea>DIR
Volume in drive C has no label.
Volume Serial Number is 008E-E804

Directory of C:\MyWorkArea

13/02/2007  09:12    <DIR>          -
13/02/2007  09:12    <DIR>          -
13/02/2007  09:11                17 FILE-ONE.TXT
13/02/2007  09:11                17 FILE-THREE.TXT
13/02/2007  09:11                17 FILE-TWO.TXT
                3 File(s)          51 bytes
                2 Dir(s)  14,227,746,816 bytes free

C:\MyWorkArea>
```

2. Put a blank formatted floppy drive in drive A:

From the command prompt (in your working directory) issue the command:

COPY FILE-ONE.TXT A: (Be sure to get the spaces between the command and its argument right and don't forget the carriage return). You will see a message 'One file(s) copied'.

3. Make the A: drive current by typing A: [+Carriage Return] at the prompt. The prompt will change to A:
4. Use the DIR command to see that the file has been copied.
5. DELete the file on the A: drive.
6. Make the C: drive current again – the prompt will change to confirm this.
7. Use the command COPY *.* A: to copy all files to the A: drive. Note the results.
8. Make the A: drive current.
9. Use the DIR command to list the files that you have copied.
10. DELete them with the command DEL *.*

When you have completed this exercise and noted the results, experiment further. Try setting some (or all) of your files to be Read Only (ATTRIB +R FILENAME.EXT to do this) and attempt to step through the exercise again. Set your files to be 'Hidden', then try to go through the exercise again. Look up the HELP facilities for the command that you have used.

Now that you have experimented with the COPY command, look up the HELP entry for its 'grown up' version, the XCOPY command, then experiment with that.

Moving and Renaming files

Rather than copying files it is sometimes useful to be able to move them to the target.

You could, of course, do this by copying them, then deleting the original. However,

you can do this in a single operation with the MOVE command. Try the previous exercise substituting MOVE for COPY and note the results.

To RENAME a file use the REN command REN THISFILE.EXT THATFILE.EXT.

You can also COPY and rename a file in a single operation. If you have made the three files suggested earlier try making a fourth with the command:

```
COPY FILE-ONE.TXT FILE-FOUR.TXT
```

Use the DIR command to verify that you now have an extra file.

Summary (2)

In this section we have seen:

- How to create and remove directories.
- How to navigate between directories and drives.
- How to create files with the COPY CON command.
- How to examine and change file attributes.
- How to delete files.
- How to copy files.
- How to move files.
- How to use wildcards in order to work with groups of files.

3. Using the EDIT.COM editor

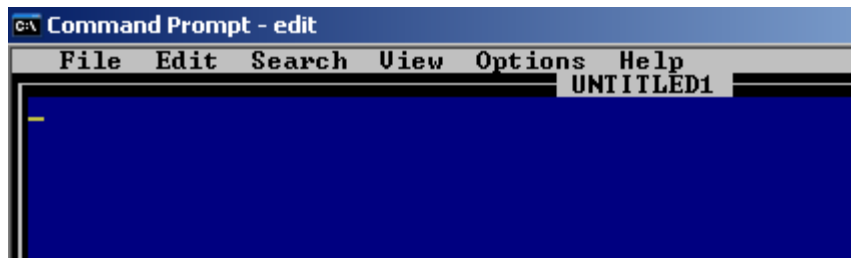
In the course of your working life as a technician you will need to be able to edit or create plain text files, particularly Batch files and CMD scripts (more later). In order to do this you can use any text editor such as Windows Notepad or even a Word Processor such as Open Office Writer which can save files in plain (ASCII) text format. However, these editors depend on having a working Graphical User Interface (GUI) and this will not always be available. If, for instance, you need to boot a system into safe mode with a command prompt, you will need to be able to use a text based

editor. Two text editors are available at the Windows command line – EDLIN.EXE, a line editor which is a legacy (more leftover, really) from the days of DOS – or EDIT.COM which is rather easier to use.

To start the EDIT editor, open a command prompt window and type EDIT at the prompt. You will see something like figure 10.

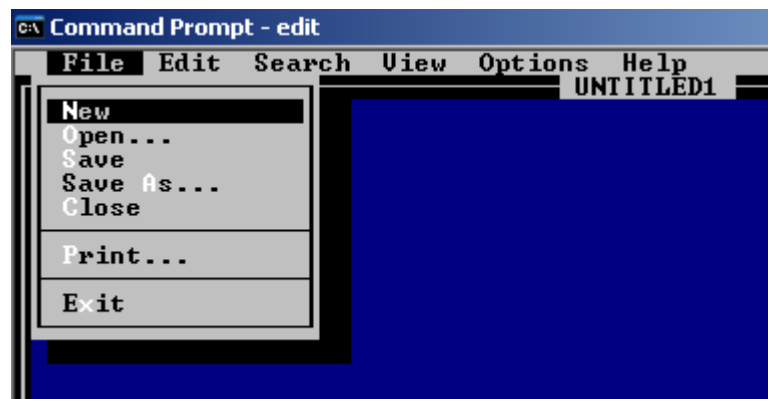
Figure 10

If you are running your command prompt from inside a GUI then you can access the



Menu items at the top of the editor window with your mouse. If, on the other hand, you are working from a prompt in DOS mode you will need to be able to manage without a mouse. In order to access a menu entry from the keyboard, hold down the Alt key at the same time as the first letter of the menu entry. So, to access the File menu hold down ALT+F – Figure 11 illustrates this.

Figure 11



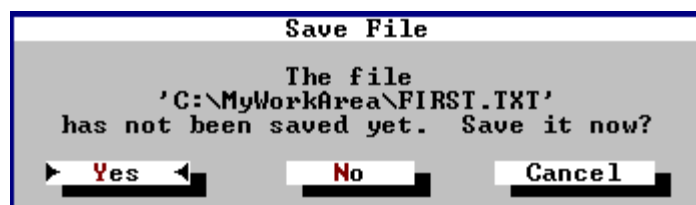
You can use the Up and Down arrow keys to move up and down the File menu and the Left and Right arrow keys to move along the menu bar. Each menu item has one highlighted letter which corresponds to the Alt+key combination needed to invoke it. So, for example, Alt+N in the File menu in figure 11 will open a new file, Alt+X will exit from the editor, and so on. If this is the first time you have used EDIT.COM it is probably worthwhile familiarising yourself with the basics of its menus and commands.

EDITing your first file

You can either start the editor with the EDIT command (without arguments) or you can provide a file name. Where you provide a file name, EDIT will open the file of that name if it exists or create a new one where it doesn't. To create a new file with EDIT called FIRST.TXT

1. Issue the command EDIT FIRST.TXT (capital letters are used here for clarity but you can use whatever you like).
2. Enter a single line of text such as 'line one of the text'.
3. Use Alt+S to save it, followed by Alt+X to close the editor – note that if you simply used Alt+X EDIT would give you the option to save your file as in figure 12

Figure 12



4. Close the editor, saving your file.

5. Start the editor with the command `EDIT FIRST.TXT`. You will see your saved text.
6. Modify your file (or not, as you choose) and familiarise yourself with the menu system and the keyboard commands needed to use it.
7. Exit.

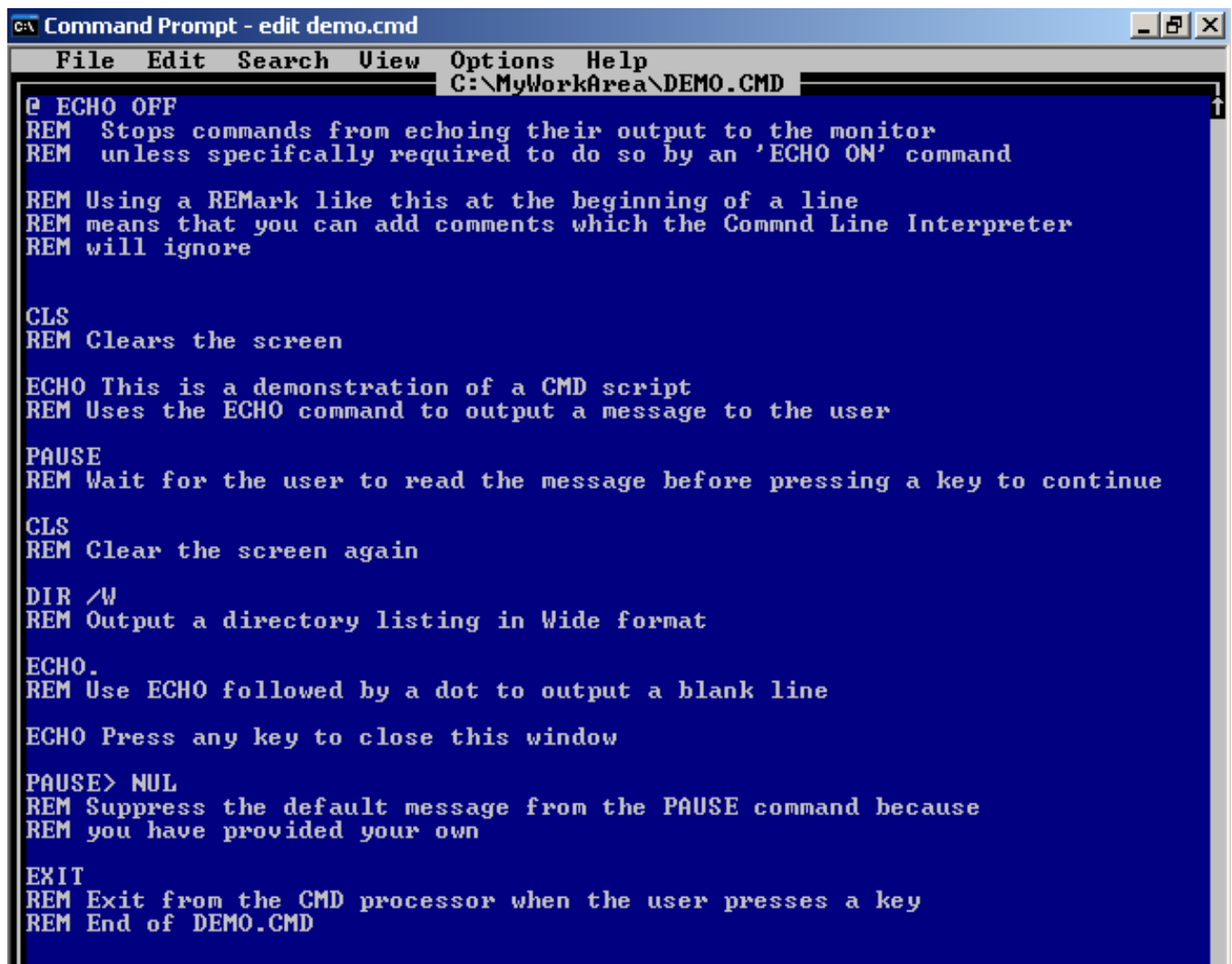
4. CMD scripts

Any command that you type at a command line can be included in a script which is run by typing its name at a command prompt. Provided that each (valid) command is on a line of its own you can include several lines. In order to be recognised by the system as a script the file name must have the extension `.CMD`. (There is a similar type of script file which runs under the legacy `COMMAND.COM` processor which has the extension `.BAT`. Windows decides which command line processor to use based on the file extension, so `BATch` files run under `COMMAND.COM` and `CMD` files under `CMD.EXE`).

In order to demonstrate how a `CMD` script works we will create a small example called `DEMO.CMD`.

1. Start `EDIT` with the command `EDIT DEMO.CMD`
2. Enter some commands, one per line, as in figure 13
3. Save your file and close the editor.
4. Run the file by typing `DEMO` at the prompt (note you don't need the `.CMD` extension to make it run, but it has to be there for the system to recognise it.)

Figure 13



```
Command Prompt - edit demo.cmd
File Edit Search View Options Help
C:\MyWorkArea\DEMO.CMD
@ ECHO OFF
REM Stops commands from echoing their output to the monitor
REM unless specifically required to do so by an 'ECHO ON' command

REM Using a REMark like this at the beginning of a line
REM means that you can add comments which the Command Line Interpreter
REM will ignore

CLS
REM Clears the screen

ECHO This is a demonstration of a CMD script
REM Uses the ECHO command to output a message to the user

PAUSE
REM Wait for the user to read the message before pressing a key to continue

CLS
REM Clear the screen again

DIR /W
REM Output a directory listing in Wide format

ECHO.
REM Use ECHO followed by a dot to output a blank line


ECHO Press any key to close this window

PAUSE > NUL
REM Suppress the default message from the PAUSE command because
REM you have provided your own

EXIT
REM Exit from the CMD processor when the user presses a key
REM End of DEMO.CMD
```

When you run this file your output will be something like that in figure 14.

Figure 14



```
Command Prompt - DEMO
Volume in drive C has no label.
Volume Serial Number is 008E-E804

Directory of C:\MYWORK~1

[.]          [..]          DEMO.CMD          FILE-ONE.TXT
FILE-THREE.TXT  FILE-TWO.TXT  FIRST.TXT
                5 File(s)          988 bytes
                2 Dir(s) 13,931,552,768 bytes free

Press any key to close this window
```

As it stands this is not a particularly useful script. However, it demonstrates the basics of how a script is put together so you can use it to devise scripts of your own which may be more useful. You could, for example, write a script which backs up files which have been modified since the last backup using the XCOPY command with appropriate switches. If you want to see some professionally written CMD scripts you can use the Windows search utility to look for *.CMD (note the use of the wild card * symbol) and take a look at them. If you do this, be careful not to modify any of the scripts as most of them are part of Windows and changing them could make your system unstable or even unusable.

Summary (3)

In the previous two sections we have looked at how to use the EDIT.COM editor to edit files and to create CMD script files. We have done this at an introductory level and, for CMD scripts in particular, there is much more that you can do. As with most things in computing, learning by doing is the most effective way of getting to grips with the material so you should spend some time writing, running and debugging your own CMD scripts.

5. Some network commands

PING

The PING command is one of those everyday tools that you will use when working on any TCP / IP network from the Internet to the office LAN. Its purpose is to check connectivity between network nodes and you can PING any remote host by name or by IP address. If you use the name of the remote device as the first argument of the PING command it will (if successful) return the IP address of the remote machine.

Figure 15 shows the result of PINGing a local file server PC called VICTORIA.

```

C:\ Command Prompt

C:\MyWorkArea>PING VICTORIA

Pinging VICTORIA [192.168.2.10] with 32 bytes of data:

Reply from 192.168.2.10: bytes=32 time<1ms TTL=64
Reply from 192.168.2.10: bytes=32 time<1ms TTL=64
Reply from 192.168.2.10: bytes=32 time<1ms TTL=64
Reply from 192.168.2.10: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\MyWorkArea>_

```

Note that PING returns the IP address of the target, the number of packets sent, the number received and the percentage lost. The TTL field is the Time To Live for each of the data packets sent. This is a field which is decremented each time it passes through a network node so that if a packet is undeliverable it will be 'killed' when these field becomes zero. This prevents undeliverable packets circulating indefinitely.

TRACERT

The TRACERT command traces the route between the system on which it is run and the remote host specified in the command. Figure 16 shows the result of a TRACERT on the Internet.

Figure 16

```

C:\ Command Prompt

C:\MyWorkArea>TRACERT WWW.BBC.CO.UK

Tracing route to www.bbc.net.uk [212.58.224.88]
over a maximum of 30 hops:

  1  <1 ms    <1 ms    <1 ms    . [192.168.2.1]
  2  22 ms    20 ms    19 ms    lns2.186k.co.uk [195.82.97.251]
  3  20 ms    22 ms    21 ms    bdr3.lon-th1.mailbox.net.uk [195.82.97.230]
  4  19 ms    117 ms   23 ms    rt-lonap-a.thdo.bbc.co.uk [193.203.5.90]
  5  20 ms    19 ms    20 ms    212.58.238.129
  6  26 ms    27 ms    21 ms    www51.thdo.bbc.co.uk [212.58.224.88]

Trace complete.

```

The output listed in the figure shows the number of 'hops' between source and target and the IP addresses of each router that it passes through. One of the practical uses of this command is determining the number of hops (and therefore the likely response


times) between a connected PC and an on-line server that is used for multi-player interactive games.

NET

NET is actually a suite of utilities. The help facility NET /? lists the commands and NET [NAME OF COMMAND] gives more detailed help. Figure 17 shows the output for the NET VIEW command.

Figure 17

Used without arguments NET VIEW will list all of the PC's on your LAN. Figure 18

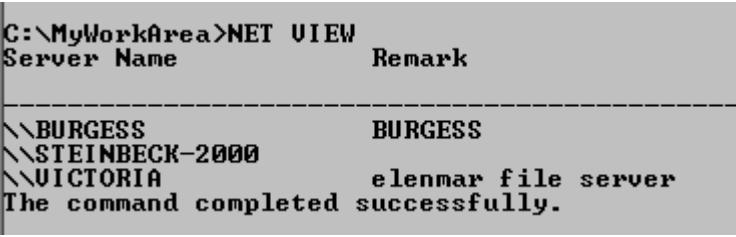


```
C:\ Command Prompt
C:\MyWorkArea>NET VIEW /?
The syntax of this command is:

NET VIEW
[\\computername [/CACHE] ; /DOMAIN[:domainname]]
NET VIEW /NETWORK:NW [\\computername]
```

shows an example of this.

Figure 18



```
C:\MyWorkArea>NET VIEW
Server Name          Remark
-----
\\BURGESS             BURGESS
\\STEINBECK-2000
\\VICTORIA            elenmar file server
The command completed successfully.
```

6. Some useful oddments

A short tutorial such as this cannot possibly cover all there is to know about command line work; it can only introduce the basics and leave it to the reader to conduct their own investigations. The following commands are simply listed because they are sometimes useful in practice.

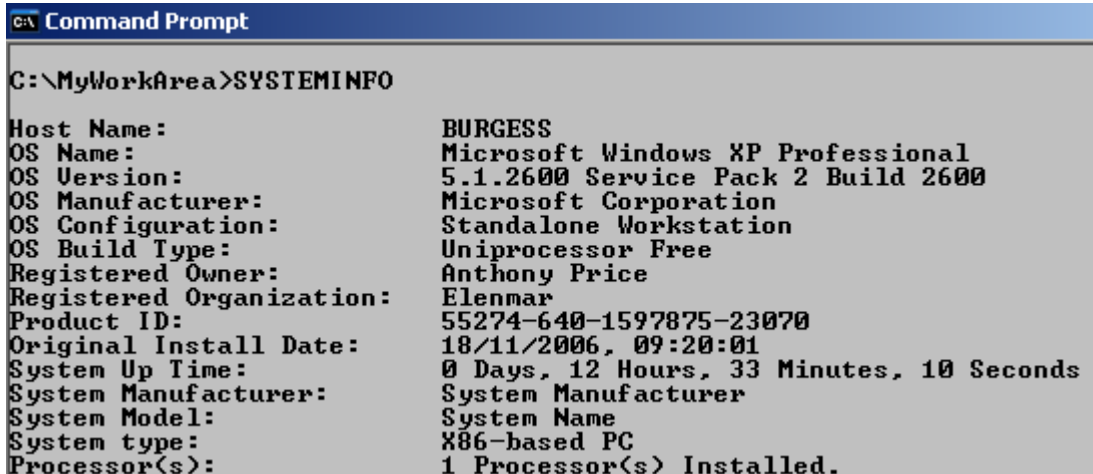
CLS

We have already seen – in the section on CMD scripts – that CLS is the command that is used to clear the screen. It can be used at a prompt simply to clear away the clutter left behind by previous commands so that you can concentrate on what you are doing. It can also remove errors arising from your mistakes so that anyone watching over your shoulder doesn't go away with the impression that you haven't got a clue!

SYSTEMINFO

The SYSTEMINFO command is not available on Windows 2000 which is a pity because it is a good source of System Information. Figure 19 shows *some* of the output of this command on a Windows XP Professional machine.

Figure 19



```
C:\MyWorkArea>SYSTEMINFO

Host Name:                BURGESS
OS Name:                  Microsoft Windows XP Professional
OS Version:               5.1.2600 Service Pack 2 Build 2600
OS Manufacturer:        Microsoft Corporation
OS Configuration:       Standalone Workstation
OS Build Type:            Uniprocessor Free
Registered Owner:        Anthony Price
Registered Organization:  Elenmar
Product ID:               55274-640-1597875-23070
Original Install Date:    18/11/2006, 09:20:01
System Up Time:           0 Days, 12 Hours, 33 Minutes, 10 Seconds
System Manufacturer:     System Manufacturer
System Model:             System Name
System type:              X86-based PC
Processor(s):             1 Processor(s) Installed.
```

This command is not part of the default install of XP Home, but some technicians copy it from an XP Professional PC and it will work though it may be an infringement of the End User Licence Agreement (EULA). This won't work on Windows 2000

MSCONFIG

This is *the* configuration utility which has been present on all Windows systems since 9.x but is not included in Windows 2000. The Windows XP version will work in

2000. It can even be run a standalone utility. Some technicians copy a version from XP to 2000 or even have a copy of the MSCONFIG.EXE program on a removable disk; it will fit easily on to a floppy along with SYSTEMINFO.EXE

UP+DOWN ARROW KEYS

When you open a command line window the operating system keeps a 'stack' of all the command that you have entered. This means that you can step forwards or backward through the command stack and this can save you some typing. Unlike a Linux command prompt which keeps the command history in a file, Windows only keeps the information for the current session in memory and this is lost when the command window is closed.

TASKLIST and TASKKILL – (XP Professional)

TASKLIST is a utility which lists all currently running tasks on an XP Professional system – the command line equivalent of the **Processes** tab of the *Windows Task Manager*. Figure 20 shows part of a TASKLIST list on an XP machine.

Figure 20

```
C:\>TASKLIST
```

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Console	0	16 K
System	4	Console	0	220 K
smss.exe	772	Console	0	372 K
csrss.exe	1028	Console	0	4,444 K
winlogon.exe	1052	Console	0	676 K
services.exe	1096	Console	0	3,384 K
lsass.exe	1108	Console	0	1,336 K
svchost.exe	1272	Console	0	4,936 K
svchost.exe	1316	Console	0	4,356 K
svchost.exe	1740	Console	0	28,856 K
svchost.exe	1784	Console	0	3,260 K
svchost.exe	1852	Console	0	4,628 K
vsmon.exe	1916	Console	0	21,116 K
spoolsv.exe	892	Console	0	4,912 K
explorer.exe	1924	Console	0	33,420 K
avgcc.exe	152	Console	0	236 K
zclclient.exe	200	Console	0	6,036 K
NetworkDriveManager.exe	208	Console	0	5,200 K
jusched.exe	232	Console	0	1,988 K
realsched.exe	1224	Console	0	144 K
mixer.exe	404	Console	0	4,516 K
E_FAT19EE.EXE	412	Console	0	2,540 K
ReplayRadio.exe	436	Console	0	4,728 K
ctfmon.exe	508	Console	0	4,116 K
HprSnap6.exe	616	Console	0	3,992 K
MailWasher.exe	636	Console	0	1,792 K
avgamsvr.exe	992	Console	0	464 K
avgupsvc.exe	1280	Console	0	1,004 K
avgemc.exe	1416	Console	0	2,016 K
SAgent2.exe	1460	Console	0	2,964 K
svchost.exe	1724	Console	0	4,120 K
vmware-authd.exe	1352	Console	0	3,556 K
vmount2.exe	1572	Console	0	3,684 K
vmnat.exe	1860	Console	0	1,896 K
vmnetdhcp.exe	1912	Console	0	1,632 K
alg.exe	3136	Console	0	3,464 K
soffice.exe	3656	Console	0	1,932 K
soffice.bin	3680	Console	0	59,416 K
cmd.exe	3968	Console	0	3,368 K
wmiprvse.exe	3536	Console	0	6,140 K
tasklist.exe	3224	Console	0	4,892 K

Each running process has an **Image Name** (so that you can identify it) and a **Process Identity (PID)** which identifies it to the operating system. The level of detail of the output can be controlled by various command line switches – TASKLIST /? will list details at the prompt.

TASKKILL provides you with a way of terminating a task from a command line. Its simplest usage is TASKKILL /PID [PID]. If, for example, you wished to terminate the instance of CMD.EXE in figure 20, the command TASKKILL /PID 3968 would do the job. As a practical demonstration, you could open a CMD window, run TASKLIST then kill off your instance of CMD. This will terminate the process, i.e.

terminate that instance of CMD and close the window where it is running. If you attempt to kill a critical system task such as WINLOGON.EXE, the system will refuse to do this and output an error message. There are other options and switches – TASKKILL /? will give details.

7. Summary

In this tutorial we have looked at some of the basics of command line work. By working through the material presented here you will learn the basic commands, how to get help on commands, how to use a command line editor and how to create and run CMD script files. More importantly you should be able to develop a feel for command line work. For example, a Recovery Console (in Windows) uses a different set of commands to CMD, and the various 'Shells' in UNIX or Linux are very different. However, if you are used to working at a CMD prompt in Windows these skills are fairly portable to other systems such as Linux. The learning curve for command line work can be fairly steep in the early part (and you may even fall off it a couple of times) but once you have learned them, command line skills are both powerful and portable in your working life as a technician.

[END]