

MS-DOS Tutorial

You should work either on your Liverpool Hope I: drive or your pen-drive. This tutorial will assume you are using the I: drive. If not, you will need to substitute the name of your own working drive wherever this tutorial refers to the I: drive.

Exercise

Obtain a copy of the zip file MS-DOS_EXERCISES.zip from the Moodle site and unzip it in the root of your I: drive. This will create the folder EXERCISES containing the sub-folders and files you will be working with in this tutorial.

1. The default drive

Exercise

Open an Command Prompt box by selecting **Start|Programs|Accessories|Command Prompt**

The Command Prompt you see on the screen might look like this:

```
C:\WINDOWS\system32>
```

Each disc drive attached to a computer system is identified by a name. In MS-DOS, the name is a single letter followed by a colon.

e.g. A:, C:

Traditionally, A: is a floppy drive while C: is a hard disc drive. They are also described as Drive A and Drive C.

The system prompt C:\WINDOWS\system32> identifies the **default drive** as drive C. Unless a command specifies otherwise, it is the default drive which is used by MS-DOS whenever a request to use a file is received.

As is usual in computing, there is alternative name for the **default drive**. It is also known as the **current drive**.

On a hard disc system, on switching on, the default drive is usually the hard disc, drive C.

To change the default drive to drive I:, we type:

I:

followed by **<Enter>**. The system prompt now changes to I:\>

All MS-DOS commands in this worksheet will be given in bold uppercase characters. In order to execute the command, we follow it by pressing **<Enter>**. Please note that although commands are given here in uppercase, you can mix upper and lower case.

Exercise

Change the default drive to I. - [or the drive letter for your pen drive]

The MS-DOS prompt has changed to I:\>

Exercise

Now change the default drive back to C ready for the next section.

The prompt should have changed back to C:\WINDOWS\system32>

2. Volumes

We can label discs magnetically as well as by external sticky labels. The disc can be given a **volume identification label** (or **volume** for short) when the disc is formatted.

To see the volume of a disc type:

VOL

Exercise

Find out the volume identification label for drive C. Next make I the default drive and examine the volume label on your disc.

Your pen drive disc may not have a volume label. We can give it one by using the command:

LABEL

This command prompts you for a name (maximum 11 characters).

Exercise

Think of a suitable volume name for your pen drive disc and give the disc this name.

3. Clearing the screen

It is always nice to start new tasks on a clear screen rather than have previous work scrolling up out of sight. The command **CLS** allows us to do this.

Exercise

Clear the screen on your computer.

4. Setting the date and time

Even while the machine is switched off, the computer maintains a **system clock** so that it can keep track of the date and time. If you need to check the date and time you can use two commands : **DATE** and **TIME**.

Using the command **DATE** results in output like

Current date is: 23/09/2009

Enter new date (dd-mm-yy):

Pressing <Enter> leaves the computer's version of the date as it is whilst typing 22-09-09 would reset the date to 22nd September 2009. Note that the prompt tells you how to present the new date.

Similarly the command **TIME** allows you to check and reset the time.

Exercise

Check and reset if necessary the time and date on your computer.

5. The directory

The command **DIR** will make a listing on the screen of the files on your disc.

Exercise

Change the default drive to drive C (by using the command **C:**).

Make a listing on the screen of files on drive C. If there are many files then you will see them scrolling off the top of the screen.

Exercise

Change the default drive back to drive I (by using the command **I:**). Now change the current drive to I:\EXERCISE by typing the command **CD \EXERCISE <Enter>**.

You should now see the new MS-DOS prompt:.

I:\EXERCISE>

Exercise

Carry out **DIR** on the disc in drive I.

You should see something like this:

Volume in drive I is MY_VOL

Volume Serial Number is F114-D4AE

Directory of I:\EXERCISE

```
23/09/2009  15:08    <DIR>          .
23/09/2009  15:08    <DIR>          ..
24/08/1996  10:11                69,886 EDIT.COM
23/09/2009  14:46    <DIR>          LETTERS
23/09/2009  14:46    <DIR>          PROGRAMS
13/12/1994  13:23                210 STUDENTS.TXT
              3 File(s)              70,096 bytes
              4 Dir(s)  80,759,676,928 bytes free
```

There are several important points about this file list:

- The first line shows the Volume label.
- The third line shows the **Directory**.
- The last column represents the **filename** (note that "filename" is just one word) of the file.
- If the file is in fact a subdirectory, then the symbol <DIR> appears in the third column.
- The first and second columns give the date and time when the file was last altered.
- The first two entries in the directory:

```
23/09/2009  15:08    <DIR>          .
23/09/2009  15:08    <DIR>          ..
```

are used by the operating system to keep track of where the directory is in relation to the other directories. You can ignore them.

6. Viewing the contents of a large directory

DIR/P causes the output to pause at the end of each screenful. The display will resume when you press any key.

DIR/W causes the file names to be displayed, five files on each line of the screen. Directory names are shown surrounded by [] brackets.

DIR C: causes the directory in drive C to be listed even though the default drive may be drive A.

Exercise

Predict the outcome of the following sequence of commands:

```
I:  
DIR C:/W  
DIR
```

Try them out to see if they match your expectations.

7. File Types

Generally you can choose whatever names you like for your files as long as you keep to the restrictions on character types and lengths of filename and extension.

However, certain extensions have specific meanings within MS-DOS

A file with the extension **COM** such as **EDIT.COM** contains a **COM** program, which is just a program written in machine code. MS-DOS can also execute **EXE** programs which are also machine code programs and are stored in files with the extension **EXE**.

COM programs are limited to 64 kbytes in size whereas for an EXE program there is no size limit.

The extension **BAT** is reserved for **batch files**. These are text files which contain MS-DOS commands. They allow you to create new commands using existing MS-DOS commands as building blocks. The filename of the batch file becomes the name of the new command.

8. Batch files

Suppose we want to invent a single command called **STARTUP** to prompt us to adjust the date and time and then clear the screen.

The file would contain:

```
DATE
TIME
CLS
```

The file would be named **STARTUP.BAT**.

When you type **STARTUP** followed by <Enter>, each of the commands listed inside it will be carried out in sequence.

9. The MS-DOS Editor

You can create STARTUP.BAT by using the command:

EDIT STARTUP.BAT

Exercise

Make sure that I: is the default drive.

Create the file STARTUP.BAT using the **EDIT** command.

This command starts the MS-DOS screen editor. Type in the three lines and then save the file using **FileSave** . Selecting **FileExit** gets you out of the editor and back to the MS-DOS prompt.

Test the **STARTUP** command to confirm that it works.

10. The TYPE command

Files containing text can be directly displayed on the screen using the **TYPE** command.

This command must be followed by the name of the file you want displayed. Not all file names have an extension. However, if a file name includes an extension, it must be included when using the **TYPE** command.

e.g. **TYPE STUDENTS.TXT**

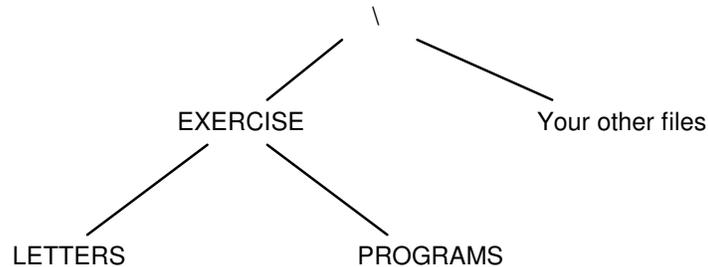
Exercise

Make sure you are in drive I and in directory EXERCISE. Use **CD \EXERCISE** to get into the directory if you are not already there..

Display the contents of the file STUDENTS.TXT on the screen

11. Hierarchical directories and paths

Here is the directory structure you added to your disc at the beginning of these notes:



The root (\) is the directory to which the system first defaults.

12. Looking down the directory structure

When you open a Command Prompt box you will be automatically put into a directory of the default drive - normally drive C. At Hope, this is usually the `WINDOWS\system32` directory on drive C. This is the meaning of the prompt

```
C:\WINDOWS\system32>
```

The C: tells you that drive C is the default drive.

If you make I: the current drive then the prompt changes to

```
I:\>
```

You are now in the root directory of the disc in drive I.

Exercise

Make sure that I is the current drive by typing **I:**

Type **CD** to make sure that the root directory is the current directory.

Use **DIR** to see the files on your disc

Among the files already on your disc you should also see this entry in the directory listing:

```
23/09/2009 15:08 <DIR> EXERCISE
```

This is the directory that you placed on your disc at the beginning of these notes.

At this moment we cannot see what is inside EXERCISE since it is not the current directory.

However, the command **DIR EXERCISE** will show us its contents.

Exercise

List on screen the files inside the directory EXERCISE.

Notice that there are two sub-directories inside EXERCISE. One of them is called LETTERS. To see what is inside the LETTERS subdirectory we can use the command **DIR EXERCISE\LETTERS**.

Exercise

List on screen the files inside the directory LETTERS which is a sub-directory of EXERCISE

In order to read the letter MAVIS1.TXT we can use the command **TYPE EXERCISE\LETTERS\MAVIS1.TXT**

Exercise

Display on the screen the contents of the love letter to Mavis

13. Changing the current directory

As you can imagine, several layers of directories will lead to very long full names for files. An alternative strategy is to **change the current directory**.

To make EXERCISE the current directory we use the command:

CD EXERCISE

CD stands for Change Directory

Exercise

Make EXERCISE the current directory

Notice that the prompt has changed to:

I:\EXERCISE>

The prompt is telling us the current directory along with the current drive.

Now that we are in the EXERCISE directory we can see the directory contents by just using the command **DIR**.

Exercise

Use **DIR** to list on screen the contents of the current directory.

Use the **CD** command again to make LETTERS the current directory.

Now the prompt has changed to:

I:\EXERCISE\LETTERS>

Now MAVIS1.TXT is in our current directory. we can see its contents by using:

TYPE MAVIS1.TXT

We can return to the root directory by using

**CD **

Instead of moving directory by directory, we could have travelled from \ to LETTERS in one move using

CD \EXERCISE\LETTERS

Exercise

Return to the root directory then move from \ to LETTERS using one command. Check your arrival with **DIR**

14. Selecting files

You can select files one at a time or use characters known as **wildcards** to select groups of files.

The beauty of consistent filenames becomes apparent when you want to do something with several files as a group (such as moving all files that have extension .DOC).

The two wildcards are

- ? which represents one valid character in this place in the name
- * which represents any number of valid characters finishing off this part of the file's name.

Suppose a directory contained the following files:

A.COM AB.COM ABB.COM B.COM A.BAT B.BAT

DIR A.COM	selects A.COM
DIR ?.COM	selects A.COM and B.COM
DIR *.COM	selects A.COM AB.COM ABB.COM and B.COM
DIR ?.BAT	selects A.BAT and B.BAT
DIR *.BAT	selects A.BAT and B.BAT
DIR A?.COM	selects A.COM and AB.COM
DIR A*.COM	selects A.COM, AB.COM and ABB.COM
DIR A.*	selects A.COM and A.BAT
DIR ?.*	selects A.COM, B.COM, A.BAT and B.BAT
DIR *.*	selects all the files.

Exercise

Make PROGRAMS the current directory.

Select:

- (a) all the BAT files
- (b) all the files with B the second letter in the name

15. Making a new directory

The command **MD** (standing for Make Directory) is used to make a new directory.

Thus the command:

MD \EXERCISE\ARCHIVE

will create a subdirectory of EXERCISE called ARCHIVE no matter where we are in the directory structure.

If EXERCISE had been the current directory we could have used

MD ARCHIVE

to obtain the same effect.

Exercise

Make ARCHIVE a new subdirectory of LETTERS. Use DIR to check that it is there.

16. Copying files

Copying files is usually a simple matter of creating a duplicate of a file in another directory. You cannot create a duplicate in the same directory without renaming the file - otherwise the operating system would not be able to distinguish between them.

Sometimes copying files to another location on your disc makes sense. Perhaps you want to place a copy of a file in an ARCHIVE directory while continuing to work on it in the current directory.

Beware though that creating multiple copies may confuse you as to which is the up to date copy. Looking at the time and date stamps in the directory listings will help.

To copy all the files in LETTERS to the new ARCHIVE directory we would need

```
COPY \EXERCISE\LETTERS\*.* \EXERCISE\LETTERS\ARCHIVE
```

If LETTERS had been the current directory we could have used instead:

```
COPY *.* ARCHIVE
```

Exercise

Copy all the files in LETTERS to ARCHIVE

Make a directory listing for ARCHIVE to confirm that the copy has in fact taken place.

17. Deleting files

Files can be removed from a directory by using the DEL command:

```
DEL \EXERCISE\PROGRAMS\A.BAT
```

removes A.BAT from sub-directory PROGRAMS.

To remove the file MINE.TXT from the current directory we use

```
DEL MINE.TXT
```

To remove all the files in the current directory we use

```
DEL *.*
```

Since removing all the files in a directory is a drastic action, MS-DOS asks you if you are sure you want to go ahead.

18. Removing directories

We can only remove directories which contain no files. Thus to remove ARCHIVE might use this sequence of commands:

```
CD \EXERCISE\LETTERS\ARCHIVE
```

```
DEL *.*
```

```
CD .. (This command moves us up to the parent directory)
```

```
RD ARCHIVE
```

Exercise

How could we do all these operations without moving from the root directory?

19. Useful Commands for Batch Files

PAUSE temporarily pauses execution of the batch file. Use this command if you need time to read a screen before moving onto the next batch command.

ECHO *message* displays *message* on the screen

ECHO OFF use this command to stop *subsequent* commands from your batch files appearing on the screen. Without this, MS-DOS will display the command line before executing it (as happened in the exercise following Section 8). **ECHO ON** resumes displaying commands.

@command instructs MS-DOS not to display the *current* command line.

For example, to prevent any command lines from being displayed, use **@ECHO OFF** as the first line in the batch file.

GOTO *label* jumps over all the subsequent lines in the batch file until it finds the line labelled *:label* . For example **GOTO END** jumps to the line labelled **:END** .

REM causes the rest of the line to be ignored. This is how you can add comments to a batch file to explain what is going on.

These commands will be demonstrated in the following sections.

20. Replaceable parameters

We can enhance batch files by using **replaceable parameters** which allow you to pass information to the batch file by including information on the command line when you run the batch file.

For example, the following batch file COPYCOMP.BAT copies all the files from the C:\STORE directory to drive I: and the compares the files:

```
@ECHO OFF
COPY C:\STORE I:
FC C:\STORE I: /L
```

Type **FC /?** to see what FC does.

Suppose you want to make the file more versatile so that you can use it to copy and compare the files in any directory. We revise COPYCOMP.BAT as follows:

```
@ECHO OFF
COPY %1 %2
FC %1 %2 /L
```

Notice that %1 and %2 replace C:\STORE and A: respectively. These parameters are replaceable parameters. You can, in fact use up to 9 parameters using the names %1 up to %9. Note that %0 holds the name of the command itself and is rarely used.

After making these changes in COPYCOMP.BAT you can use the batch file to copy and compare the files from any directory to any disc or directory. To copy and compare the files in the \MYWORK\LETTERS directory on drive C: with the files on drive B:, for example, type the following command:

```
COPYCOMP C:\MYWORK\LETTERS B:
```

Exercise

Write a batch file with a replaceable parameter to implement a command with the following syntax:

CURRDIR *directory*

The command makes *directory* the current directory then performs a directory listing with 5 file names on each line of the screen.

21. The IF command

MS-DOS provides some handy decision-making capabilities for batch files by means of the IF command. The capabilities of the IF command can be grouped into 3 categories:

- **IF EXIST** [*path*]*filename command* causes MS-DOS to execute *command* if that file exists.

e.g.

```
@ECHO OFF
IF EXIST \WORK\MYFILE GOTO DONE
ECHO The file does not exist
:DONE
```

- **IF string1 == string2** *command* causes MS-DOS to execute *command* if string1 and string2 are identical.

```
@ECHO OFF
IF '%1' == '' GOTO TRY_AGAIN
IF '%1' == 'W' GOTO RUN_WORD
IF '%1' == 'S' GOTO RUN_SPREAD
:TRY_AGAIN
    ECHO Try again! You must use W or S
    GOTO END
:RUN_WORD
    "C:\PROGRAM FILES\MICROSOFT OFFICE\OFFICE11\WINWORD.EXE"
    GOTO END
:RUN_SPREAD
    "C:\PROGRAM FILES\MICROSOFT OFFICE\OFFICE11\EXCEL.EXE"
:END
```

S

Note that string1 and string2 must be inside quotes - as well as %1 and %2

We can use NOT in IF commands

e.g. `IF NOT EXIST myfile.txt ECHO It is not there`

22. Pausing for input in a batch file

We can pause to receive input from the keyboard by using the SET command with the /P switch. The command is used as follows:

`SET /P CHOICE=Your choice ?`

`IF NOT '%CHOICE%'==" SET CHOICE=%CHOICE:~0,1%`

Here are the explanations of the components of this command:

- `/P` switch allows you to set the value of a variable to a line of input entered by the user. It displays the string following the = sign as a prompt; In this case the variable is CHOICE and the prompt is "Your choice?"
- `IF NOT '%CHOICE%'=="` is comparing the contents of the variable CHOICE with an empty string - yes "" are two separate single quotes.
- `SET CHOICE==%CHOICE:~0,1%` means set the variable CHOICE to be the string formed by 1 character starting at position 0 of CHOICE's current contents. This is a strange syntax. Welcome to the weird world of computing!

In writing your batch file, you respond to the key pressed by using a series of IF commands.

23. Running batch files from other batch files

Method 1: Shifting control permanently to another batch file

Running batch files from within batch files is particularly useful if you want to create a menu batch file that can start several different programs. All you have to do is include the name of the second batch file (the .BAT is not needed but the path **is** if it is in a different directory) as a line in the first batch file. However, once the second batch file has done executing, control does **not** pass back to the first batch file. To illustrate this, suppose we had the following two batch files:

```
REM BATCH1.BAT
@ECHO OFF
ECHO This is BATCH1
BATCH2
ECHO Back to BATCH1
```

```
REM BATCH2.BAT
@ECHO OFF
ECHO This is BATCH2
```

The output is:

```
This is BATCH1
This is BATCH2
```

Method 2: Calling a batch file and returning using CALL

We can get to the original file if we use CALL. Modifying the last example:

```
REM BATCH1.BAT
@ECHO OFF
ECHO This is BATCH1
CALL BATCH2
ECHO Back to BATCH1
```

```
REM BATCH2.BAT
@ECHO OFF
ECHO This is BATCH2
```

The output is:

```
This is BATCH1
This is BATCH2
Back to BATCH1
```

24. Creating a simple menu

Here is a simple menu file.

```
REM MENU.BAT
@ECHO OFF
CLS
:START
ECHO;
ECHO;
ECHO           Menu
ECHO           ====
ECHO      D.  Set the date
ECHO      T.  Set the time
ECHO      S.  See the current directory
ECHO      Q.  Quit
ECHO;

SET /P CHOICE=Your choice ?
IF NOT '%CHOICE%'==' ' SET CHOICE=%CHOICE:~0,1%
IF '%CHOICE%'=='D' GOTO CHOICE_DATE
IF '%CHOICE%'=='T' GOTO CHOICE_TIME
IF '%CHOICE%'=='S' GOTO CHOICE_DIR
IF '%CHOICE%'=='Q' GOTO END
ECHO "%CHOICE%" is not valid. Please try again.
ECHO.
GOTO START
:CHOICE_DATE
    ECHO Loading the date program
    DATE
    GOTO END
:CHOICE_TIME
    ECHO Loading the time program
    TIME
    GOTO END
:CHOICE_DIR
    ECHO Loading the DIR program
    DIR
:END
```

When you type the MENU command, MS-DOS clears the screen and display the following:

```
           Menu
           ====
      D.  Set the date
      T.  Set the time
      S.  See the current directory
      Q.  Quit
```

Your choice ?

Index

	—B—			—H—	
batch files		6	Hierarchical directories		8
	—C—			—I—	
CALL.		18	IF 16		
CD		10	IF EXIST		16
Changing the current directory		10		—M—	
Clearing the screen		3			
CLS		3	Making a new directory		12
COM		6	MD		12
Copying files		12		—P—	
current drive		1			
	—D—		paths		8
DATE		3	PAUSE		14
default drive		1		—R—	
DIR		3			
DIR/P		5	REM		14
DIR/W		5	Removing directories		14
directory structure		8	replaceable parameters		15
	—E—			—S—	
ECHO		14	Selecting files		11
ECHO OFF		14	SET		17
ECHO ON		14		—T—	
Editor		7			
EXE		6	TIME		3
	—F—		TYPE		7
FC		15		—V—	
filename		4	Volumes		2
	—G—				
GOTO		14			