

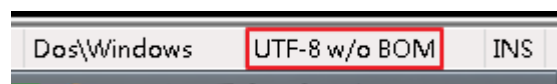
<https://peterju.gitbooks.io/cmdddoc/content/chapter1.html>

1. 批次檔介紹

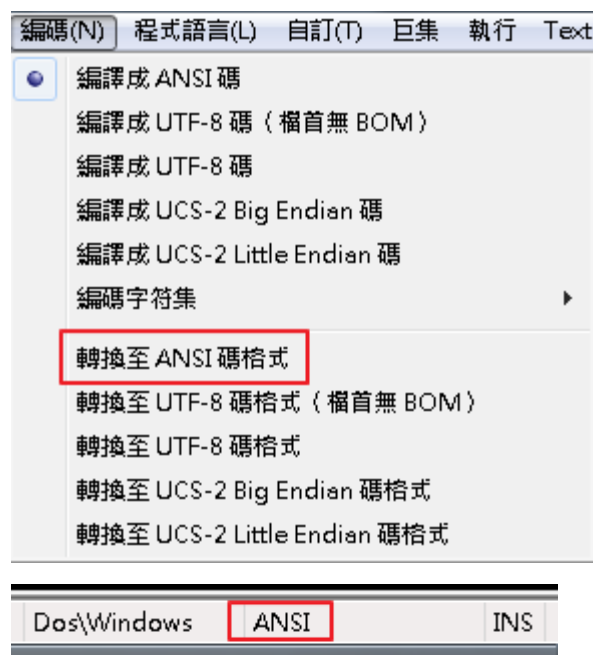
將命令提示字元(Command Prompt)中輸入的指令集結起來，輸入在文字檔中，用以批次執行，稱之為批次(Batch file)檔。

批次檔指令每行的長度預設為 127 個字元，執行後若要中斷可按下 Ctrl+C。

命令提示字元預設的字碼頁為 ANSI/BIG5 編碼，因此檢視 UTF-8 編碼檔案時會出現亂碼，所以批次檔的編寫應盡量使用 ANSI 的編碼方式，建議使用 [NotePad++](#) 之類的有顏色與語法提示的純文字編輯器編寫。NotePad++ 目前編輯文字檔採用的編碼方式會在右下角提示，



建議透過功能表:編碼 / 轉換至 ANSI 編碼格式，將編碼轉換為 ANSI 格式。



學習批次檔等同於學習一個作業系統在命令列環境(shell)的用法，Windows 環境目前有很多 Linux 的影子，例如導向、管線、指令與檔案名稱補齊...等，理解之後未來接觸 Linux 也有很大的幫助。

學習批次檔一般而言相當的簡單又直覺，怎樣算是學會呢，以本人的經驗，至少要將變數延遲展開特性與 for 迴圈指令給透徹了解才算是掌握了批次檔喔。

副檔名

預設有下面這 2 種，在 DOS 與 Windows 9x 時代副檔名為.bat，在 Windows NT 之後則改用.cmd，表示在視窗模式下的命令提示字元(cmd.exe)執行

- bat
- cmd

註解方式

標準是使用 rem，但通常會使用 2 個以上的冒號來當註解符號，畢竟字數比較少嘛

- rem
- ::

顯示訊息

批次檔中用來顯示訊息的指令是 echo，其後可加上字串或變數(可混搭)，在正常的情况下，批次檔中的每道指令執行前都會先出現螢幕上，使用 echo off 指令，就可以關閉顯示指令，通常在不需要互動的批次檔中都一定會出現。

```
@ech off
set /P myname=Please input your name:
echo Hello %myname%
echo.
echo Today is %date% %time%
pause
```

- 停止下達的指令顯示在螢幕上 => echo off
- 空一行 => echo.
- 顯示當前目錄 => echo %cd%
- 顯示日期 => echo %date%
- 顯示時間 => echo %time%

判斷(if)

if /? 說明寫的不錯，就請直接參考囉

暫停

- pause

程式暫停，提示按任意鍵繼續

```
echo This Program is running...
pause
```

- timeout 秒數

程式暫停指定的秒數

```
echo Please wait for a while...
timeout 6
```

回傳值

無論是在 Linux 下撰寫 shell script 或是在 Windows 下撰寫批次檔，最近一次程式執行的回傳值判斷，在撰寫工作自動化的 Script 檔時，是非常重要的技巧。

在 Windows 環境中的慣例是，若指令成功時傳回 0，若錯誤時，依據錯誤的狀況會傳回 1 或以上的值，代表不同的錯誤狀況。

但並非所有指令都會根據正確或錯誤而有不同的回傳值，要用來判斷前請先測試一下。

```
ping 168.95.192.1
echo %errorlevel%
```

```
ping 123.123.123.123
echo %errorlevel%
```

判斷回傳值錯誤的方式有 2 種

1. 判斷錯誤等級變數 if %errorlevel%==1

2. ping 123.123.123.123 > nul
3. if %errorlevel%==1 echo %date%-%time% ping command fail >> pingTarget.log
4. 錯誤判斷子句 if errorlevel 1

if errorlevel 1 表示若回傳值大於等於 1(>=1)，當使用多個錯誤判斷子句時，一定要根據 errorlevel 遞減的順序來排列，因為錯誤判斷子句有上述 2 個隱含意義，因此建議使用第 1 種判斷錯誤等級變數的方式為佳

```
ping 123.123.123.123 > nul
if errorlevel 1 goto ONE
if errorlevel 0 goto ZERO
goto END
:ZERO
echo %date%-%time% ping command succesful >> pingTarget.log
goto END
:ONE
echo %date%-%time% ping command fail >> pingTarget.log
:END
```

命令列參數

假設在命令列鍵入了下列指令

```
test.cmd c:\windows\notepad.exe c:\windows\write.exe
```

此時批次檔內部自動將命令列上的參數視為特別的變數如下表

%0 (命令)	\$1 (參數 1)	%2 (參數 2)
test.cmd	c:\windows\notepad.exe	c:\windows\write.exe

請以 Notepad++ 編輯 test.cmd 並鍵入下面的指令，存檔後在命令列上輸入

```
test.cmd c:\windows\notepad.exe c:\windows\write.exe
```

```
@ech off
echo command = %0
echo argument1 = %1
echo argument2 = %2
```

擴充字元參數

承上例，若在批次檔內的命令列參數加上擴充字元之後，可額外得知參數的許多資訊，請參考下表

%1	%~d1 (取得磁 碟機代號)	%~p1 (取得路 徑)	%~n1 (取得 檔名)	%~x1 (取得 副檔名)
c:\windows\notepad.exe c:		\Windows\	notepad	.exe
echo %~d1				
echo %~p1				
echo %~n1				
echo %~x1				

更多的擴充字元請輸入 `call/?` 或參考 <http://inpega.blogspot.tw/2012/07/cd-dp0.html> 來獲得。

跳行與結束程式

- goto 標籤 (須定義標籤，標籤須單獨一行，並以冒號開頭)
- if not exist c:\temp\nul goto newdir
- cd /d c:\temp
- del * /y
- :newdir
- mkdir c:\temp
- goto :eof (無須定義標籤，直接結束程式之意)
- set /p end=按 0 結束程式:
- if %end%==0 goto :eof
- exit /b [回傳值]

使用 `exit /b` 可停止批次檔或副程式的執行，若結束後需要提供回傳值讓 `if` 指令 檢查 `errorlevel` 變數，可於其後加上想要的數值

```
rem myping.cmd 168.95.192.1
@echo off
ping %1 > nul
if %errorlevel%==1 goto error
ping %1 ok
```

```
exit /b 0
:error
echo ping fail
exit /b 1
```

設定副程式結束後回傳值為 5

```
@echo Off
call :setError
echo %errorlevel%
goto :eof
```

```
:setError
Exit /B 5
```