1.Echo 命令

打开回显或关闭请求回显功能，或显示消息。如果没有任何参数，echo 命令将显示当前回显设置。

语法:

echo [{on│off}] [message]  
Sample：@echo off / echo hello world

在实际应用中我们会把这条命令和重定向符号（也称为管道符号，一般用> >> ^）结合来实现输入一些命

令到特定格式的文件中.这将在以后的例子中体现出来。

2.@ 命令

表示不显示@后面的命令，在入侵过程中（例如使用批处理来格式化敌人的硬盘）自然不能让对方看到你使

用的命令啦。

Sample：@echo off  
@echo Now initializing the program,please wait a minite…  
@format X: /q/u/autoset (format 这个命令是不可以使用/y这个参数的，可喜的是微软留了个autoset这

个参数给我们，效果和/y是一样的。)

3.Goto 命令

指定跳转到标签，找到标签后，程序将处理从下一行开始的命令。

语法：goto label （label是参数，指定所要转向的批处理程序中的行。）  
Sample：  
if {%1}=={} goto noparms  
if {%2}=={} goto noparms（如果这里的if、%1、%2你不明白的话，先跳过去，后面会有详细的解释。）  
@Rem check parameters if null show usage  
:noparms  
echo Usage: monitor.bat ServerIP PortNumber  
goto end

标签的名字可以随便起，但是最好是有意义的字母啦，字母前加个：用来表示这个字母是标签，goto命令

就是根据这个：来寻找下一步跳到到那里。最好有一些说明这样你别人看起来才会理解你的意图啊。

4.Rem 命令

注释命令，在C语言中相当与/\*——–\*/,它并不会被执行，只是起一个注释的作用，便于别人[阅读](http://book.2cto.com/" \t "http://www.2cto.com/kf/200610/_blank)和你

自己日后修改。

Rem Message  
Sample：@Rem Here is the description.

5.Pause 命令

运行 Pause 命令时，将显示下面的消息：  
Press any key to continue . . .

Sample：  
@echo off  
:begin  
copy a:\*.\* d：ack  
echo Please put a new disk into driver A  
pause  
goto begin

在这个例子中，驱动器 A 中磁盘上的所有文件均复制到d:ack中。显示的注释提示您将另一张磁盘放入

驱动器 A 时，pause 命令会使程序挂起，以便您更换磁盘，然后按任意键继续处理。

6.Call 命令

从一个批处理程序调用另一个批处理程序，并且不终止父批处理程序。call 命令接受用作调用目标的标签

。如果在脚本或批处理文件外使用 Call，它将不会在命令行起作用。

语法:  
call [[Drive:][Path] FileName [BatchParameters]] [:label [arguments]]

参数:  
[Drive:}[Path] FileName

指定要调用的批处理程序的位置和名称。filename 参数必须具有 .bat 或 .cmd 扩展名。

7.start 命令

调用外部程序，所有的DOS命令和命令行程序都可以由start命令来调用。  
常用参数：  
MIN 开始时窗口最小化  
SEPARATE 在分开的空间内开始 16 位 Windows 程序  
HIGH 在 HIGH 优先级类别开始应用程序  
REALTIME 在 REALTIME 优先级类别开始应用程序  
WAIT 启动应用程序并等候它结束  
parameters 这些为传送到命令/程序的参数

执行的应用程序是 32-位 GUI 应用程序时，CMD.EXE不等应用程序终止就返回命令提示。如果在命令脚本

内执行，该新行为则不会发生。

8.choice 命令

choice 使用此命令可以让用户输入一个字符，从而运行不同的命令。使用时应该加/c:参数，c:后应写提

示可输入的字符，之间无空格。它的返回码为1234……

如: choice /c:dme defrag,mem,end

将显示  
defrag,mem,end[D,M,E]?

Sample：  
Sample.bat的内容如下:

@echo off  
choice /c:dme defrag,mem,end  
if errorlevel 3 goto defrag （应先判断数值最高的错误码）  
if errorlevel 2 goto mem  
if errotlevel 1 goto end

:defrag  
c:[dos](http://www.2cto.com/os/dos/" \t "http://www.2cto.com/kf/200610/_blank)defrag  
goto end  
:mem  
mem  
goto end  
:end  
echo good bye

此文件运行后，将显示 defrag,mem,end[D,M,E]? 用户可选择d m e ，然后if语句将作出判断，d表示执行

标号为defrag的程序段，m表示执行标号为mem的程序段，e表示执行标号为end的程序段，每个程序段最后

都以goto end将程序跳到end标号处，然后程序将显示good bye，文件结束。

9.If 命令

if 表示将判断是否符合规定的条件，从而决定执行不同的命令。

有三种格式:

1)、if “参数” == “字符串” 　待执行的命令  
参数如果等于指定的字符串，则条件成立，运行命令，否则运行下一句。(注意是两个等号）

如if “%1″==”a” format a:  
if {%1}=={} goto noparms  
if {%2}=={} goto noparms

2)、if exist 文件名　待执行的命令  
如果有指定的文件，则条件成立，运行命令，否则运行下一句。

如if exist config.sys edit config.sys

3)、if errorlevel / if not errorlevel 数字　 待执行的命令  
如果返回码等于指定的数字，则条件成立，运行命令，否则运行下一句。

如if errorlevel 2 goto x2

DOS程序运行时都会返回一个数字给DOS，称为错误码errorlevel或称返回码，常见的返回码为0、1。

10.for 命令

for 命令是一个比较复杂的命令，主要用于参数在指定的范围内循环执行命令。  
在批处理文件中使用 FOR 命令时，指定变量请使用 %%variable

for {%variable│%%variable} in (set) do command [ CommandLineOptions]  
%variable 指定一个单一字母可替换的参数。  
(set) 指定一个或一组文件。可以使用通配符。  
command 指定对每个文件执行的命令。  
command-parameters 为特定命令指定参数或命令行开关。

在批处理文件中使用 FOR 命令时，指定变量请使用 %%variable  
而不要用 %variable。变量名称是区分大小写的，所以 %i 不同于 %I

如果命令扩展名被启用，下列额外的 FOR 命令格式会受到支持:  
FOR /D %variable IN (set) DO command [command-parameters]

如果集中包含通配符，则指定与目录名匹配，而不与文件名匹配。

FOR /R [[drive:]path] %variable IN (set) DO command [command-parameters]

检查以 [drive:]path 为根的目录树，指向每个目录中的FOR 语句。如果在 /R 后没有指定目录，则使用

当前目录。如果集仅为一个单点(.)字符，则枚举该目录树。

FOR /L %variable IN (start,step,end) DO command [command-parameters]

该集表示以增量形式从开始到结束的一个数字序列。  
因此，(1,1,5) 将产生序列 1 2 3 4 5，(5,-1,1) 将产生  
序列 (5 4 3 2 1)。

FOR /F [”options”] %variable IN (file-set) DO command  
FOR /F [”options”] %variable IN (”string”) DO command  
FOR /F [”options”] %variable IN (’command’) DO command

或者，如果有 usebackq 选项:

FOR /F [”options”] %variable IN (file-set) DO command  
FOR /F [”options”] %variable IN (”string”) DO command  
FOR /F [”options”] %variable IN (’command’) DO command

filenameset 为一个或多个文件名。继续到 filenameset 中的下一个文件之前，每份文件都已被打开、读

取并经过处理。  
处理包括读取文件，将其分成一行行的文字，然后将每行解析成零或更多的符号。然后用已找到的符号字

符串变量值调用 For 循环。以默认方式，/F 通过每个文件的每一行中分开的第一个空白符号。跳过空白

行。您可通过指定可选 “options” 参数替代默认解析操作。这个带引号的字符串包括一个或多个指定不同

解析选项的关键字。这些关键字为:

eol=c - 指一个行注释字符的结尾(就一个)  
skip=n - 指在文件开始时忽略的行数。  
delims=xxx - 指分隔符集。这个替换了空格和跳格键的默认分隔符集。  
tokens=x,y,m-n - 指每行的哪一个符号被传递到每个迭代的 for 本身。这会导致额外变量名称的  
格式为一个范围。通过 nth 符号指定 m 符号字符串中的最后一个字符星号，那么额外的变量将在最后一

个符号解析之分配并接受行的保留文本。  
usebackq - 指定新语法已在下类情况中使用:  
在作为命令执行一个后引号的字符串并且引号字符为文字字符串命令并允许在 file-set中使用双引号扩起

文件名称。

sample1:  
FOR /F “eol=; tokens=2,3\* delims=, ” %i in (myfile.txt) do command

会分析 myfile.txt 中的每一行，忽略以分号打头的那些行，将每行中的第二个和第三个符号传递给 for

程序体；用逗号和/或 空格定界符号。请注意，这个 for 程序体的语句引用 %i 来取得第二个符号，引用

%j 来取得第三个符号，引用 %k来取得第三个符号后的所有剩余符号。对于带有空格的文件名，您需要用

双引号将文件名括起来。为了用这种方式来使用双引号，您还需要使用 usebackq 选项，否则，双引号会

被理解成是用作定义某个要分析的字符串的。

%i 专门在 for 语句中得到说明，%j 和 %k 是通过tokens= 选项专门得到说明的。您可以通过 tokens=

一行指定最多 26 个符号，只要不试图说明一个高于字母 ‘z’ 或’Z’ 的变量。请记住，FOR 变量是单一字

母、分大小写和全局的；同时不能有 52 个以上都在使用中。

您还可以在相邻字符串上使用 FOR /F 分析逻辑；方法是，用单引号将括号之间的 filenameset 括起来。

这样，该字符