Windows Batch [精华]

http://en.wikibooks.org/wiki/Windows_Batch_Scripting

1. 综述

1、".bat": 这是微软的第一个批处理文件的后缀名，在几乎所有的 Windows 操作系统内都能运行。

2、 ".cmd": 是为 Windows NT 设计的命令行脚本， 为 Cmd.exe shell 而设计的， 对于 COMMAND.COM 不具有向后兼容性。

3、目前所知道的.cmd 和 .bat 文件的区别是对 ERRORLEVEL 变量的改变：当 Command Extensions （命令行扩展）处于 enabled 状态时，哪怕是.cmd 文件中一个成功执行的命令都可以改变 ERRORLEVEL 的值，而在.bat 文件中 ERRORLEVEL 变量只有在遇到错误的时候才发生改变。

2. @echo off/echo on

1. "echo" 用来在控制台上显示信息。
2. "echo." 用来显示一空行。（注意：echo 和右下角的点之间没有空格，如果有空格就变成显示句点了。）
3. "echo off": 在使用了这行命令之后，其他命令都只显示命令的结果，而不显示命令本身。
4. "echo on": 这是默认值，表示显示所有的命令结果和命令行本身。
5. "echo": 当执行 echo 而不带任何参数的时候，会显示 echo 的打开或关闭的状态："ECHO is on" 或者 "ECHO is off"。
6. "@":@ 符号表示不显示本行的命令本身。如果只用 echo off，虽然 echo off 后面的命令不显示出来，只显示命令的结果，但是 echo off 它自己确被显示出来了，这就是使用@echo off 的原因。

3. SETLOCAL/ENDLOCAL

1. SETLOCAL 用来控制批处理文件中变量的可见性。就是高级语言常说的局部变量。凡是在 SETLOCAL 和 ENDLOCAL 之间的变量都是局部的，以免被其他脚本文件改变变量的值，而没有使用这个标示的都是 Global visible（全局变量），很可能被其他文件所改变。下面的例子很好的说明了这一点。

@echo off setlocal set version=1.0 echo the first version is %version% endlocal echo the second version is %version% ::The follow is global variable set version=2.0 echo the third version is %version%	执行这个文件，将输出： the first version is 1.0 the second version is the third version is 2.0 第二个 version 是全局变量，但是没有定义，所以是空值。再次执行： the first version is 1.0 the second version is 2.0 the third version is 2.0
@echo off	执行第二个文件：

setlocal set version=1.0 echo the first version is %version% endlocal echo the second version is %version% ::The follow is global variable set version=5.0 echo the third version is %version%	the first version is 1.0 the second version is 2.0 the third version is 5.0 再次执行： the first version is 1.0 the second version is 5.0 the third version is 5.0

由此可见变量的作用域在简单的批处理文件还是很慎重，否则可以造成千变万化的结果，这样这个脚本就很不稳定了。所以，变量作用域很重要，得注意。

2. EnableDelayedExpansion. （参考百度：http://baike.baidu.com/view/2923132.htm ）

代码	结果
@echo off setlocal set var=test & echo show %var% endlocal	show 当解析到%var%是，CMD 还没有执行完任何语句，所以系统并不认识这个变量，所以认为变量没有定义，所以显示空值。
@echo off Setlocal ENABLEDELAYEDEXPANSION set var=test & echo show !var! endlocal	show test 使用了变量延迟扩展，并使用!var!，其实在执行的时候也不认识这个变量，但是这条语句执行完以后，仍然返回去赋值。这种事后处理，就是变量延迟扩展。
For 语句看起来写了很多行，其实认为是一条语句。所以如果在没有使用变量延迟扩展的时候，在 For 语句里的每一行，都只是记忆着系统所记忆的东西。而使用了变量延迟扩展之后，会执行完了整条语句，再“事后处理”给与赋值，然后才显示到控制台。下面的例子很能说明问题。	
@echo off setlocal :: count to 5 set _tst=0 FOR /l %%G in (1,1,5) Do (echo [%_tst%] & set /a _tst+=1) echo Total = %_tst%	[0] [0] [0] [0] [0] Total = 5
@echo off setlocal :: count to 5 set _tst=0 FOR /l %%G in (1,1,5) Do (echo [!_tst!] & set /a _tst+=1) echo Total = !_tst!	[0] [1] [2] [3] [4] Total = 5

在 CMD 里面，系统使用“^”作为脱字符。使用变量延迟扩展 ENABLEDELAYEDEXPANSION 之后，脱字符^ 就可以一直产生作用，而不再局限于一个命令行。这样对于处理包含 HTML 和 XML 格式的字符的时候，就方便多了。请看下面的例子。

```
@echo off
REM 这个文件会创建 World.txt 到本地磁盘，文件内容为 Hello。
REM 也就是第二次运行的时候，脱字符没有发挥作用。
SETLOCAL
Set _html= Hello^>World.txt
Echo %_html%
ENDLOCAL
```

```
@echo off
REM 这个文件会输出 Hello>World.txt，脱字符一直产生作用。
SETLOCAL EnableDelayedExpansion
Set _html= Hello^>World.txt
Echo !_html!
ENDLOCAL
```

```
@echo off
REM 使用变量延迟扩展来处理 html 语句。
SETLOCAL EnableDelayedExpansion
Set _html=html format: ^<title^>Hello world^</title^>
Echo !_html!
ENDLOCAL
```

3. DISABLEEXTENSIONS.

值得注意的是，这里的扩展不是扩展名，而是命令的扩展功能，比如“/F”参数就是 For 循环的命令扩展。

只有使用了命令扩展我们才能执行下面的语句。

```
FOR /F "DELIMS=" %%a IN (test.TXT) DO @echo %%a
```

否则我们要输出一个文件，只能使用 Type test.txt 了。命令扩展默认是开启的。

4. Set

1. SET 不带任何参数：将显示当前用户的所有系统环境变量。(set)
2. SET 带一个变量名：尝试搜索并显示以这个变量名开头的环境变量。（set PROCESSOR 就会显示以 PROCESSOR 开始的变量）
3. SET variable=string：给一个变量赋值。

4. SET “”: 使用一对空的双引号，会显示一个不带参数的情况下没显示出来的变量。我这里就显示下面两个奇怪的变量。

   =C:=C:\Users\Lingli

   =E:=E:\Powershell

   使用 cd %=C:%还真可以转到 C:\Users\Lingli 目录下。

5. SET “var=”(or SET var=): 删除一个变量。
6. SET /A variable=expression: 使用算数运算符来给变量赋值。

```
()              - 组合
! ~ -           - 一元操作符
* / %           - 算数运算符
+ -             -算数运算符
<< >>           - 逻辑偏移
&               - 位与
^               - 位异或
|               - 位或
= *= /= %= += -=    - 赋值
 &= ^= |= <<= >>=
```

7. SET /P variable=[提示字符]:提示用户输入并把输入的值赋给 variable。提示字符可以为空。有时可以使用 CHOICE 来代替 SET /P.

```
@echo off
setlocal
set /p version=Please enter the QQ version:
echo you will install QQ %version%
endlocal
```

8. 有用的环境变量。

```
%CD% - 当前路径名。
%DATE% - 当前日期。
%TIME% - 当前时间。
%RANDOM% - 显示 0 到 32767 之间的一个随机数。你看 CMD 也是可以获取随机数的。
```

5. Choice

1. 详细语法：

CHOICE [/C choices] [/N] [/CS] [/T timeout /D choice] [/M text]

该命令可以提供用户一个选择列表，并使用 ERRORLEVEL 参数返回用户选择项的序号。第一项对于 1，第二项对应 2，依次类推，如果用户按 CTRL+C 退出而不选择，就返回 0。

/C choices	用字母列出所提供的选项，默认是"YN"，即是和否。

/N	隐藏选择列表。如果使用这个选项，那么提示的文字里最后写清楚以下。不推荐使用。
/CS	激活大小写敏感。默认是对大小写不敏感。
/T timeout	选择超时设置。
/D choice	超时后的默认选项。
/M text	提示字符。最好把要选择的问题将清楚。

2. 例子:

```
@echo off
SETLOCAL
CHOICE /C ABC /M "选择登陆用户：A - 管理员；B 域用户；C 注册用户"
IF %errorlevel%==1 goto :Admin
IF %errorlevel%==2 goto :Domain
IF %errorlevel%==3 goto :Local
goto Exit

:Admin
    echo 欢迎管理员
    goto Exit

:Domain
    echo 欢迎域用户
    goto Exit

:Local
    echo 欢迎注册用户
    goto Exit

:Exit
endlocal
```

6. Rem and Arguments.

1. REM [comment]: 批处理文件的注释符，可以使用“::” 来代替 REM。
2. 参数。

%0 对应文件名本身，其他分别对应一个参数值，支持 255 个参数。

Tes	1	2	3		r		2
%0	%	%	%		%		%

3. 文件名参数扩展。

当使用文件名作参数的时候，可以使用如下文件名扩展。下面的例子扩展%1，其实对所有的参数都可以进行类似的扩展。

%~f1 - 扩展%1 为带路径的全名。
%~d1 - 只显示磁盘名称。
%~p1 - 只显示文件路径。
%~n1 - 只显示文件名，不包括后缀名，也不包括路径。
%~x1 - 只显示后缀名。
%~s1 - 变成短文件名，将包含“~”符号。
%~1 - 有时候文件名包含空格是，会对文件名加双引号。这个功能是去掉双引号。
%~a1 - 显示文件的属性。
%~t1 - 显示文件的修改时间。
%~z1 - 显示文件的大小。

上面的扩展是可以组合的:
%~dp1 - 扩展%1 为磁盘名称和路径名。
%~nx2 - 扩展%2 为文件名和文件后缀名。

```
::Test.bat
::Example: test.bat test.bat
@echo off
setlocal
set fn=%~f1
echo %fn%
endlocal
```

7. IF 语句

1. 详细语法：

File syntax

```
IF [NOT] EXIST filename command
IF [NOT] EXIST filename (command) ELSE (command)
```

```
String syntax
        IF [/I] [NOT] item1==item2 command
        IF [/I] item1 compare-op item2 command
        IF [/I] item1 compare-op item2 (command) ELSE (command)

Error Check Syntax
        IF [NOT] DEFINED variable command
        IF [NOT] ERRORLEVEL number command
        IF CMDEXTVERSION number command

Key
        /I          : 比较时不考虑文件名大小写，即大小写不敏感。
        compare-op : 逻辑比较
        EQU : equal 等于
        NEQ : not equal 不等于
        LSS : less than 小于
        LEQ : less than or equal 小于或等于
        GTR : greater than 大于
        GEQ : greater than or equal 大于或等于
```

2. 例子(可以看出 If 语句也是一条语句)

```
IF EXIST filename (del filename) ELSE ( echo The file was not found.)
```

```
IF EXIST filename (
del filename
) ELSE (
echo The file was not found.
)
```

8. For 循环

1. 详细语法：

```
FOR-Files
        FOR %%parameter IN (set) DO command

FOR-Files-Rooted at Path
        FOR /R [[drive:]path] %%parameter IN (set) DO command

FOR-Folders
        FOR /D %%parameter IN (folder_set) DO command

FOR-List of numbers
        FOR /L %%parameter IN (start,step,end) DO command
```

```
FOR-File contents
        FOR /F ["options"] %%parameter IN (filenameset) DO command
        FOR /F ["options"] %%parameter IN ("Text string to process") DO
            command

FOR-Command Results
             FOR /F ["options"] %%parameter IN ('command to process')
            DO command



        eol=c           - 行首注释符。如果有这个注释符，这一行不做处理。
        skip=n          - 制定前 n 行跳过，不处理
        delims=xxx      - 分隔符集合。默认分隔符是空格和 Tab。
        tokens=x,y,m-n - 选择哪一块用于循环操作。
        usebackq        - 当文件名有空格需要使用双引号时，使用 usebackq 把双
            引号内的东西当文件处理，否则当字符串处理。

        注意:
        在批处理文件中，参数使用%%parameter，而当你拿到命令行使用的时
            候，使用%parameter，就是一个百分号的区别。同时变量名区分大
            小写，所以%g 和 G 是不相同的。
        (set) 说明是文件集合，一个文件或多个文件，还可以对文件名使用通配符。
```

2. 例子：

```
@echo off
setlocal
for %%G in (*.bat *.txt) do echo %%G
endlocal
```

9. Net.

1. 管理服务: Net start, stop, pause, continue [service]。
2. 连接到一个共享文件：Net use。
   NET USE [磁盘名:] \\共享名称[\子目录名] [密码] [/USER:[域名\]用户名]
   NET USE 磁盘名:]: /delete
3. Net share: 显示本地的所有共享，包括隐藏的共享。
4. Net share ShareName: 显示共享的相关信息。
5. 创建一个本地共享: NET SHARE sharename=drive:path /REMARK:"text" [/CACHE:Manual | Automatic | No]
6. 修改用户数量限制和标示。

NET SHARE sharename /USERS:number /REMARK:"text"
NET SHARE sharename /UNLIMITED /REMARK:"text"

7. 删除共享: NET SHARE {sharename | devicename | drive:path} /DELETE
8. Net view \\计算机名: 列出远程机器的所有共享。
9. Net localgroup: 把一个账户加入一个本地组，如加入管理员组：net localgroup administrators DomainName\UserName /add
10. 机器重命名: **netdom** renamecomputer 原机器名 /newname:修改后的机器名 /UserD:user /PasswordD:password
11. 加入域: net dom join 计算机名/domain:域名 /UserD:域管理员账户 /PasswordD:域管理员密码