

bat 批处理切换到当前脚本所在文件夹

切换到当前脚本所在的文件夹

```
1 | cd %~dp0
```

另外附上一些bat基本内容

批处理常用命令总结 - 批处理命令简介

目录

echo

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。

```
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的文件中。

rem

注释命令，类似于在C语言中的/*-----*/，它并不会被执行，只是起一个注释的作用，便于别人阅读和你自己日后修改。

Rem Message

Sample : @Rem Here is the description.

pause

暂停命令。运行 Pause 命令时，将显示下面的消息：
Press any key to continue. . . (或：请按任意键继续. . .)

Sample :

```
@echo off
```

公告

昵称：愤怒的企鹅

园龄：5年8个月

粉丝：0

关注：5

+加关注

<		2018年8月						>
日	一	二	三	四	五	六		
29	30	31	1	2	3	4		
5	6	7	8	9	10	11		
12	13	14	15	16	17	18		
19	20	21	22	23	24	25		
26	27	28	29	30	31	1		
2	3	4	5	6	7	8		

搜索

找找看

谷歌搜索

常用链接

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

我的标签

[linux\(23\)](#)

[boot\(17\)](#)

[docker\(11\)](#)

[c/c++\(6\)](#)

[go\(6\)](#)

[python\(6\)](#)

[tesseract\(4\)](#)

[vs qt\(4\)](#)

[web\(3\)](#)

[kali\(3\)](#)

[更多](#)

随笔档案

[2018年5月 \(1\)](#)

[2018年4月 \(2\)](#)

:begin

copy G:*. * d : /back

echo 请插入另一张光盘...

pause

goto begin

在这个例子中，驱动器 G 中磁盘上的所有文件均复制到d:/back中。显示的注释提示您将另一张光盘放入驱动器 G 时，pause 命令会使程序挂起，以便您更换光盘，然后按任意键继续处理。

call

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。

```
call [[Drive:][Path] FileName ] [:label  
[arguments]]
```

参数

[Drive:}[Path] FileName

指定要调用的批处理程序的位置和名称。

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。如：start calc.exe 即可打开Windows的计算器。

- 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。

2018年3月 (9)

2018年1月 (3)

2017年12月 (27)

2017年11月 (16)

2017年9月 (1)

2017年8月 (3)

阅读排行榜

1. bat 批处理切换到当前脚本所在文件夹(2917)
2. cordova linux 安装并编出第一个demo-android(1760)
3. kali 安装最新firefox的悲惨经历(1383)
4. python error: curl: (1) Protocol "https" not supported or disabled in libcurl(1198)
5. python 切换工作路径 为指定文件夹(1084)

goto

跳转命令。程序指针跳转到指定的标签，从标签后的第一条命令开始继续执行批处理程序。

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。

set

显示、设置或删除变量。

显示变量：set 或 set s 前者显示批处理当前已定义的所有变量及其值，后者显示所有以s开头的变量及值。

设置变量：set aa=abcd 此句命令便可向变量aa赋值abcd。

如果变量aa已被定义，则aa的值被修改为abcd；若aa尚未定义，则此句命令即可定义新的变量aa，同时为变量aa赋予初始值abcd。

删除变量：set aa= 此句命令即可删除变量aa。若变量aa已被定义，则删除变量aa；若aa尚未定义，则此句命令为实质意义。

需要说明的是，批处理中的变量是不区分类型的，不需要像C语言中的变量那样还要区分int、float、char

等。比如执行set aa=345后，变量aa的值既可以被
视为数字345，也可以被视为字符串345。

set命令具有扩展功能，如用作交互输入、字符串处理、数值计算等，属于高级命令范畴。

批处理常用命令及用法大全

阅读下面文字需要一定的dos基础概念，象：盘符、文件、目录(文件夹)、子目录、根目录、当前目录

每个命令的完整说明请加 /? 参数参考微软的帮助文档可以看到，在 /? 帮助里，"命令扩展名"一词会经常出现"命令扩展名"是指相对于win98的dos版本而言，每个命令新增的功能命令测试环境
win2000proSP4win98的命令功能太少，就不作研究了。

注：如果对某一命令还不是很熟悉，可以在命令行窗口下输入：命令名/?的方式来获得帮助。

例如：对dir命令的应用不熟悉，可以在命令行窗口下输入：dir /?

- echo 和 @回显控制命令
 - @ #关闭单行回显
 - echo off #从下一行开始关闭回显
 - @echo off #从本行开始关闭回显。
- 一般批处理第一行都是这个
- echo on #从下一行开始打开回显
 - echo #显示当前是 echo off 状态还是 echo on 状态
 - echo. #输出一个"回车换行"，一般就是指空白行
 - echo hello world #输出hello world

"关闭回显"是指运行批处理文件时，不显示文件里的
每条命令，只显示运行结果批处理开始和结束时，系统都会自动打开

- errorlevel程序返回码

echo %errorlevel%每个命令运行结束，可以用这个
命令行格式查看返回码用于判断刚才的命令是否执行
成功默认值为0，

一般命令执行出错会设 errorlevel 为1

- dir显示目录中的文件和子目录列表

- `dir` #显示当前目录中的文件和子目录
- `dir /a` #显示当前目录中的文件和子目录，包括隐藏文件和系统文件
- `dir c: /a:d` #显示 C 盘当前目录中的目录
- `dir c:/ /a:-d` #显示 C 盘根目录中的文件
`dir d:/mp3 /b/p` #逐屏显示 d:/mp3 目录里的文件，只显示文件名，不显示时间和大小
- `dir *.exe /s` #显示当前目录和子目录里所有的.exe文件其中 * 是通配符，代表所有的文件名，还有一个通配符？代表一个
- #任意字母或汉字如 `c*.*` 代表以 c 开头的所有文件？.exe 代表所有文件名是一个字母的.exe文件如果指定的目录或文件
- #不存在，将返回 errorlevel 为1;
-
- # 每个文件夹的 `dir` 输出都会有2个子目录 . 和 ... 代表当前目录.. 代表当前目录的上级目录
- `dir .` #显示当前目录中的文件和子目录
- `dir ..` #显示当前目录的上级目录中的文件和子目录

其它参数可参考 `dir /?`

- `cd`更改当前目录
- `cd mp3` #进入当前目录中的mp3目录
- `cd ..` #进入当前目录中的上级目录
- `cd/` #进入根目录
- `cd` #显示当前目录
- `cd /d d:/mp3` #可以同时更改盘符和目录
- `cd "Documents and Settings"/All users`
文件名带空格，路径前需要加上引号！！如果更改到的目录不存在，则出错返
- `md abc` #在当前目录里建立子目录 abc
- `md d:/a/b/c` #如果 d:/a 不存在，将会自动创建

- `rd abc` #删除当前目录里的 abc 子目录，要求为空目录
- `rd /s/q d:/temp` #删除 d:/temp 文件夹及其子文件夹和文件，不需要按 Y 确认
- `del d:/test.txt` #删除指定文件，不能是隐藏、系统、只读文件
- `del *.*`删除当前目录里的所有文件，不包括隐藏、系统、只读文件，要求按 Y 确认
- `del /q/a/f d:/temp/*.*`删除 d:/temp 文件夹里面的所有文件，包括隐藏、只读、系统文件，不包括子目录
- `del /q/a/f/s d:/temp/*.*`删除 d:/temp 及子文件夹里面的所有文件，包括隐藏、只读、系统文件，不包括子目录
- `ren 1.txt 2.bak` #把 1.txt 更名为 2.bak
- `ren *.txt *.ini` #把当前目录里所有.txt 文件改成.ini文件
- `ren d:/temp tmp` #支持对文件夹的重命名
- `type c:/boot.ini` #显示指定文件的内容，程序文件一般会显示乱码
- `type *.txt` #显示当前目录里所有.txt 文件的内容
- `copy c:/test.txt d:/`复制 c:/test.txt 文件到 d:/
- `copy c:/test.txt d:/test.bak`复制 c:/test.txt 文件到 d:/，并重命名为 test.bak
- `copy c:/ *.*`复制 c:/ 所有文件到当前目录，不包括隐藏文件和系统文件不指定目标路径，则默认目标路径为当前目录
- `copy con test.txt`从屏幕上等待输入，按 Ctrl+Z 结束输入，输入内容存为test.txt文件 con代表屏幕，prn代表打印机，nul代表空设备
- `copy 1.txt + 2.txt 3.txt`合并 1.txt 和 2.txt 的内容，保存为 3.txt 文件如果不指定 3.txt，则保存到 1.txt
- `copy test.txt +`复制文件到自己，实际上是修改了文件日期
- `title 新标题` #可以看到cmd窗口的标题栏变了
- `14 label` 和 `vol`设置卷标

- vol #显示卷标
- label #显示卷标，同时提示输入

新卷标

- label c:system #设置C盘的卷标为system
- md创建目录
- rd删除目录
- del删除文件
- ren文件重命名
- cls清屏
- type显示文件内容
- copy拷贝文件
- title设置cmd窗口的标题
- ver显示系统版本
- pause暂停命令

运行该命令时，将显示消息：请按任意键继续...，
一般用于看清楚屏幕上显示的内容

- rem 和 ::注释命令

注释行不执行操作

- date 和 time日期和时间
- date #显示当前日期，并提示输入新日期，按"回车"略过输入
- date/t #只显示当前日期，不提示输入新日期
- time #显示当前时间，并提示输入新时间，按"回车"略过输入
- time/t #只显示当前时间，不提示输入新时间
- :label #行首为:表示该行是标签行，标签行不执行操作
- goto label #跳转到指定的标签那一行
- find "abc" c:/test.txt在 c:/test.txt 文件里查找含 abc 字符串的行如果找不到，将set errorlevel 返回码为1
- find /i "abc" c:/test.txt查找含 abc 的行，忽略大小写
- find /c "abc" c:/test.txt显示含 abc 的行的行数
- more c:/test.txt #逐屏显示 c:/test.txt 的文件内容

- `tree d:/` #显示D盘的文件目录结构
- `c: & cd/ & dir /w`相当于把下面3行命令写到1行去了:`cd/dir /w`
- `f: && cd/ && dir >c:/test.txt`注意如果f盘不存在,那么后面2条命令将不会执行
- `find "ok" c:/test.txt && echo 成功`如果找到了"ok"字样,就显示"成功",找不到就不显示
- `find "ok" c:/test.txt || echo 不成功`如果找不到"ok"字样,就显示"不成功",找到了就不显示
- `goto` 和 `:`跳转命令
- `find` (外部命令)查找命令
- `more` (外部命令)逐屏显示
- `tree`显示目录结构
- `&`顺序执行多条命令,而不管命令是否执行成功
- `&&`顺序执行多条命令,当碰到执行出错的命令后将不执行后面的命令
- `||`顺序执行多条命令,当碰到执行正确的命令后将不执行后面的命令
- `|`管道命令

前一个命令的执行结果输出到后一个命令

`dir *.* /s/a | find /c ".exe"`管道命令表示先执行`dir`命令,对其输出的结果执行后面的`find`命令该命令行结果:

输出当前文件夹及所有子文件夹里的.exe文件的个数

`type c:/test.txt|more`这个和`more c:/test.txt`的效果是一样的

- `>` 和 `>>`输出重定向命令

`>` 清除文件中原有的内容后再写入

`>>` 追加内容到文件末尾,而不会清除原有的内容
 主要将本来显示在屏幕上的内容输出到指定文件中指定文件如果不存在,则自动生成该文件

`echo hello world>c:/test.txt` 生成`c:/test.txt`文件,内容为`hello world`这个格式在批处理文件里用得很多,可以生成`.reg` `.bat` `.vbs` 等临时文件

`type c:/test.txt >prn` 屏幕上不显示文件内容,转向输出到打印机

echo hello world>con 在屏幕上显示hello world，实际上所有输出都是默认 >con 的

copy c:/test.txt f: >nul 拷贝文件，并且不显示"文件复制成功"的提示信息，但如果f盘不存在，还是会显示出错信息

copy c:/test.txt f: >nul 2>nul 不显示"文件复制成功"的提示信息，并且f盘不存在的话，也不显示错误提示信息

echo ^^W ^> ^W>c:/test.txt 生成的文件内容为 ^W > W^ 和 > 是控制命令，要把它们输出到文件，必须在前面加个 ^符号

- < 从文件中获得输入信息，而不是从屏幕上

一般用于 date time label 等需要等待输入的命令，这样就可以不等待输入直接修改当前日期

```
@echo off
```

```
echo 2005-05-01>temp.txt
```

```
date <temp.txt
```

```
del temp.txt
```

- %0 %1 %2 %3 %4 %5 %6 %7 %8
%9 %*命令行传递给批处理的参数
 - %0 批处理文件本身
 - %1 第一个参数
 - %9 第九个参数
 - %* 从第一个参数开始的所有参数

在C盘根目录新建test.bat，内容如下：

```
@echo off
```

```
echo %0
```

```
echo %1
```

```
echo %2
```

```
echo %*
```

运行cmd，输入 c:/test.bat "/a" /b /c /d,可以看出每个参数的含意 修改test.bat内容如下

```
@echo off
```

```
echo %1
```

```
echo %~1
```

```
echo %0
```

```
echo %~f0
```

```
echo %~d0
```

```
echo %~p0
```

```
echo %~n0
```

```
echo %~x0
```

```
echo %~s0
```

```
echo %~a0
```

```
echo %~t0
```

```
echo %~z0
```

再运行cmd，输入 `c:/test.bat "/a" /b /c /d`可以参照 `call/?` 或 `for/?` 看出每个参数的含意，注意这里可以对文件进行日期比较和大小比较：

```
echo load "%~1" "%~2">c:/test.txt
```

生成的文件内容为 `load "%1" "%2"`批处理文件里，用这个格式把命令行参数输出到文件

- **set设置变量**

引用变量可在变量名前后加 `%`，即 `%变量名%`

```
set                #显示目前所有可用的变量，包括系统变量和自定义的变量
```

```
echo %SystemDrive%  #显示系统盘盘符。系统变量可以直接引用
```

```
set p              #显示所有以p开头的变量，要是  
一个也没有就设errorlevel=1
```

```
set p=aa1bb1aa2bb2  #设置变量p，并赋值为  
= 后面的字符串，即aa1bb1aa2bb2
```

```
echo %p%           #显示变量p代表的字符串，  
即aa1bb1aa2bb2
```

```
echo %p:~6%         #显示变量p中第6个字符以  
后的所有字符，即aa2bb2
```

```
echo %p:~6,3%       #显示第6个字符以后的3  
个字符，即aa2
```

echo %p:~0,3% #显示前3个字符，即aa1

echo %p:~-2% #显示最后面的2个字符，
即b2

echo %p:~0,-2% #显示除了最后2个字符以
外的其它字符，即aa1bb1aa2b

echo %p:aa=c% #用c替换变量p中所有的
aa，即显示c1bb1c2bb2

echo %p:aa=% #将变量p中的所有aa字符
串替换为空，即显示1bb12bb2

echo %p:*bb=c% #第一个bb及其之前的所有
字符被替换为c，即显示c1aa2bb2

set p=%p:*bb=c% #设置变量p，赋值为
%p:*bb=c% ，即c1aa2bb2

set /a p=39 #设置p为数值型变量，值为
39

set /a p=39/10 #支持运算符，有小数时用去
尾法，39/10=3.9，去尾得3，p=3set /a p=p/10

 #用 /a 参数时，在 = 后面的变量
可以不加%直接引用set /a p="1&0"

 #"与"运算，要加引号。其它支持的
运算符参见set/?

set p= #取消p变量

set /p p= 请输入屏幕上显示"请输入"，并
会将输入的字符串赋值给变量p注意这条可以用来取代
choice 命令

注意变量在 if 和 for 的复合语句里是一次性全部替换
的，如

@echo off

set p=aaa

if %p%==aaa (

echo %p%

set p=bbb

echo %p%)

结果将显示aaaaaa因为在读取 if 语句时已经将所有 %p% 替换为aaa这里的"替换", 在 /? 帮助里就是指"扩充"、"环境变量扩充"可以启用"延缓环境变量扩充", 用 ! 来引用变量, 即!变量名!

```
@echo off
```

```
SETLOCAL ENABLEDELAYEDEXPANSION
```

```
set p=aaaif %p%==aaa (
```

```
echo %p%
```

```
set p=bbb
```

```
echo !p! )
```

```
ENDLOCAL
```

结果将显示aaabbb 还有几个动态变量, 运行 set 看不到

%CD% #代表当前目录的字符串

%DATE% #当前日期

%TIME% #当前时间

%RANDOM% #随机整数, 介于
0~32767

%ERRORLEVEL% #当前ERRORLEVEL 值

%CMDEXTVERSION% #当前命令处理器扩展
名版本号

%CMDCMDLINE% #调用命令处理器的原始
命令行可以用echo命令查看每个变量值, 如 echo
%time%

注意 %time%精确到毫秒, 在批处理需要延时处理时
可以用到

- start批处理中调用外部程序的命令, 否则等外部程序完成后才继续执行剩下的指令
- start explorer d:/调用图形界面打开D盘
- @echo off
- cd /d %~dp0
- regedit /s 劲舞团.reg
- start patcher.exe

不加 start 命令的话, "劲舞团"运行时, 后面会有个黑乎乎的cmd窗口

- call批处理中调用另外一个批处理的命令, 否则剩下的批处理指令将不会被执行有时有的应用程序用start调用出错的, 也可以call调用
- choice (外部命令)选择命令让用户输入一个字符, 从而选择运行不同的命令, 返回码errorlevel为1234.....

win98里是choice.com, win2000pro里没有, 可以从win98里拷过来, win2003里是choice.exe

choice /N /C y /T 5 /D y>nul延时5秒

下面是个 choice 语句的例子

@echo off

rem 以下在win2000pro运行通过, 从win98里拷的chioce.com文件

choice /c:abc aaa,bbb,ccc

if errorlevel 3 goto ccc

if %errorlevel%==2 goto bbb

if errorlevel==1 goto aaa

rem 必须先判断数值高的返回码rem 可以看到errorlevel 值的判断有3种写法, 有时某种写法不好用, 可以用另外的写法

rem 直接运行

chioce相当于运行

choice /c:yn:aaa

echo aaa

goto end

:bbb

echo bbb

goto end

:ccc

echo ccc

goto end

:end

- assoc 和 ftype文件关联assoc 设置'文件扩展名'关联, 关联到'文件类型'ftype 设置'文件类型'关联, 关联到'执行程序 and 参数'当你双击一个.txt文件时, windows并不是根据.txt直接判断用 notepad.exe 打开而是先判断.txt属于 txtfile '文件类型'再调用 txtfile 关联的命令行 txtfile=%SystemRoot%/system32/NOTEPAD.EXE %1可以在"文件夹选项"→"文件类型"里修改这2种关联

- assoc #显示所有'文件扩展名'关联
- assoc .txt #显示.txt代表的'文件类型', 结果显示 .txt=txtfile
- assoc .doc #显示.doc代表的'文件类型', 结果显示 .doc=Word.Document.8
- assoc .exe #显示.exe代表的'文件类型', 结果显示 .exe=exefile
- ftype #显示所有'文件类型'关联
- ftype exefile #显示exefile类型关联的命令行, 结果显示 exefile="%1" %* assoc .txt=Word.Document.8设置.txt为word类型的文档, 可以看到.txt文件的图标都变了
- assoc .txt=txtfile恢复.txt的正确关联 ftype exefile="%1" %*恢复 exefile 的正确关联

如果该关联已经被破坏, 可以运行 command.com , 再输入这条命令

- pushd 和 popd切换当前目录
- @echo off
- c: & cd/ & md mp3 #在 C:/ 建立 mp3 文件夹
- md d:/mp4 #在 D:/ 建立 mp4 文件夹
- cd /d d:/mp4 #更改当前目录为 d:/mp4
- pushd c:/mp3 #保存当前目录, 并切换当前目录为 c:/mp3
- popd #恢复当前目录为刚才保存的 d:/mp4一般用处不大, 在当前目录名不

确定时，会有点帮助

- subst z: //server/d #这样输入z:就可以访问//server/d了
- subst z: /d #取消该映射
- subst #显示目前所有的映射
- subst (外部命令)映射磁盘。
- xcopy (外部命令)文件拷贝

xcopy d:/mp3 e:/mp3 /s/e/i/y

复制 d:/mp3 文件夹、所有子文件夹和文件到 e:/，覆盖已有文件，加 /i 表示如果 e:/ 没有 mp3 文件夹就自动新建一个，否则会有询问

- 39 一些不常用的内部命令
 - >& 将一个句柄的输出写入到另一个句柄的输入中
 - <& 从一个句柄读取输入并将其写入到另一个句柄输出中
 - shift 命令行传递给批处理的参数不止9个时，用以切换参数
 - color 设置cmd窗口的显示颜色
 - prompt 更改命令提示符号，默认都是盘符:/路径/>，如 c:/>
- format (外部命令)格式化硬盘

format c: /q/u/autotest/，q表示快速格式化，/autotest表示自动格式化，不需要按 Y 确认，/u表示每字节用F6覆盖硬盘数据，使其不可用软件恢复

format c: /c格式化C盘，并检测坏道

- fdisk (外部命令)硬盘分区win2000不带该命令win98里的fdisk不支持80G以上大硬盘，winme里的支持fdisk/mbr重建硬盘分区表，一般用于清除引导区病毒、还原精灵注意使用该命令不能从硬盘启动，必须软驱或光驱启动后直接运行
- ping (外部命令)

ping -l 65500 -t 192.168.1.200不停的向192.168.1.200计算机发送大小为65500byte的数据包

ping -n 10 127.0.0.1>nulping自己10次，可用于批处理延时10秒

- SC (外部命令)服务控制命令
- sc create aaa displayname= bbb start= auto binpath= "C:/WINDOWS/System32/alg.exe"创建服务，服务名称aaa，显示名称bbb，启动类型:自动可执行文件的路径"C:/WINDOWS/System32/alg.exe"
- sc description aaa "ccc"更改aaa的描述为ccc
- sc config aaa start= disabled binpath= "C:/WINDOWS/System32/svchost.exe -k netsvcs"更改aaa的启动类型:已禁用更改aaa的可执行文件的路径"C:/WINDOWS/System32/svchost.exe -k netsvcs"
- sc config aaa start= demand displayname= ddd更改aaa的启动类型:手动更改aaa的显示名称ddd
- sc start aaa启动aaa服务
- sc stop aaa停止aaa服务
- sc delete aaa删除aaa服务

批处理bat命令--获取当前盘符和当前目录和上级目录

批处理命令获取当前盘符和当前目录

%~d0 是当前盘符

%cd% 是当前目录

可以用echo %cd%进行打印测试

以下例子是命令行编译Visual Studio编写的程序：

```
@echo off
```

```
set b=%cd% //将当前目录保存到参数b中,等号前后不要有空格
```

```
C:
```

```
cd program files
```

```
cd microsoft visual studio
```

```
cd common
```

```
cd msdev98
```

```
cd bin
```

```
msdev "%b%\test.dsp" /MAKE "test- Win32
```



```
Release" /REBUILD //(VC6.0)
devenv "%b%\Tool Utility\Tool Utility.sln"
/REBUILD //(VS2010)
pause
```

注意:如果路径中包含空格,则应该将路径加文件名用双引号括起来。

出处:

<http://blog.csdn.net/yilmee/article/details/6735907>

```
=====
=====
=====
```

废话不多说,直接看例子:

```
@echo off
echo 当前盘符: %~d0
echo 当前盘符和路径: %~dp0
echo 当前盘符和路径的短文件名格式: %~sdp0
echo 当前批处理全路径: %~f0
echo 当前CMD默认目录: %cd%
pause
http://www.2cto.com/kf/201104/88450.html
=====
=====
=====
```

通过批处理取当前目录的上一级目录

```
@echo off
if %cd%==%cd:~,3% echo 当前目录已经
是%cd:~,1%盘的根目录! &goto end
cd..
set "bd=%cd%"
cd..
set "bbd=%cd%"
if "%bbd%"=="%bd%" (echo 上一级目录是:
%cd:~,1%盘的根目录!
) else call echo 上一级目录是:
"%%bd:%bbd%\=%%"
:end
pause
```

踩坑之路多回顾,不要在一个坑掉两次! THINK
TWICE, CODE ONCE!

标签: [bat](#)

好文要顶

关注我

收藏该文



愤怒的企鹅

[关注 - 5](#)

[粉丝 - 0](#)

[+加关注](#)

0

0

« 上一篇: [python 切换工作路径 为指定文件夹](#)

» 下一篇: [延迟环境变量扩展 \(bat \)](#)

posted @ 2017-12-13 16:27 愤怒的企鹅 阅读

(2917) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，访问网站首页。

【推荐】超50万VC++源码：大型组态工控、电力仿真CAD与GIS源码库！

【前端】SpreadJS表格控件，可嵌入应用开发的在线Excel

【推荐】如何快速搭建人工智能应用？



最新IT新闻:

- Google Maps分享地理位置会显示电量情况
- 科学家发现对宝宝使用“嘟嘟”等拟声叠词能够助其学习语言
- 帮公司采购中支付宝4888元红包 老板：抵扣工资
- 福布斯精英榜闹笑话 入榜贝米钱包CEO被取保候审
- FF广州更名恒大法拉第未来 恒大副总裁成新法定代

表人

» 更多新闻...



最新知识库文章:

- 成为一个有目标的学习者
 - 历史转折中的“杭派工程师”
 - 如何提高代码质量？
 - 在腾讯的八年，我的职业思考
 - 为什么我离开了管理岗位
- » 更多知识库文章...

Copyright ©2018 愤怒的企鹅