批处理脚本之For命令

2025-03-01T15:13:59+08:00

## 前言

在我们工作中，如果使用Windows系统，善用批处理命令，特别是在批量的文件处理，文本处理时能帮助我们极大地提升工作效率，起到事半功倍的效果！

但很多同学，对批处理的使用更多还停留在可以将多个command命令组合到一起执行，省去重复敲命令和等待的时间。这个其实只是批处理脚本最基本的使用方法

但要充分发挥批处理的作用，批处理脚本中的 for 命令其实才是核心，它作为一个兼顾循环处理，和文本列处理的命令工具，在提升批处理脚本效率方面，无出其右！

但 for 命令的使用，相对比较复杂，很多同学可能并未很好掌握，所以本文，我们带来关于批处理中这个核心命令的系统介绍

文中还有一个秋草发现的彩蛋，微软竟然有一个官方陈年Bug一直没有修 😂😂😂

## For 命令介绍

下面我们就来详细介绍这个 windows 批处理中的重磅命令

### **基础语法**

FOR %variable IN (set) DO command [command-parameters]

* **%variable**: for循环变量，批处理脚本中需用 %%，命令行直接执行用 %
* (**set)**： 可以是文件、目录、字符串或命令输出。。
* **command**： 指定对每组循环执行的命令。
* **command-parameters**： 控制循环行为（如 /R, /D, /L, /F 等）

如遍历集合:

@echo off  
for %%a in (苹果 香蕉 "西瓜 芒果") do echo 水果：%%a

**输出**：

水果：苹果  
水果：香蕉  
水果："西瓜 芒果"

* **说明**：默认按空格分隔元素，带引号的字符串视为整体。

### \*\* /R参数：递归遍历文件\*\*

这个参数，用于递归遍历目录及其子目录中的文件。

@echo off  
for /R "C:\Chengxiaqiucao" %%a in (\*.txt) do echo 找到文件：%%a

**功能**： - 遍历 C:\Chengxiaqiucao 及其子目录下的所有 .txt 文件。 - %%a 会包含完整文件路径（如 C:\Chengxiaqiucao\test\file.txt）。

### **/D：处理目录**

这个参数会仅遍历目录（不处理文件）。

@echo off  
for /D %%a in ("C:\\*") do echo 目录：%%a

**功能**： - 列出 C:\ 下的所有一级目录（如 Program Files, Users 等）。

#### **4. /L：生成数字序列**

这个参数和我们编程时控制循环次数比较类似。

**语法**：for /L %%a in (起始, 步长, 结束)

如下例生成等差数列：

@echo off  
for /L %%a in (1, 2, 10) do echo 当前数字：%%a

**输出**：

当前数字：1  
当前数字：3  
当前数字：5  
当前数字：7  
当前数字：9

### **/F：高级文本解析（核心参数）**

这个参数可以说是 For 命令最核心的参数，For命令大展身手，大部分情况下都是这个参数的功劳。

这个参数主要用于逐行解析文本、字符串或命令输出，并支持复杂规则。

##### **语法**：

for /F ["选项"] %%a in (数据源) do 命令

* **数据源**：文件、字符串或命令输出（用反引号 ` 包裹）。
* **选项**：delims, tokens, skip, eol, usebackq 等。

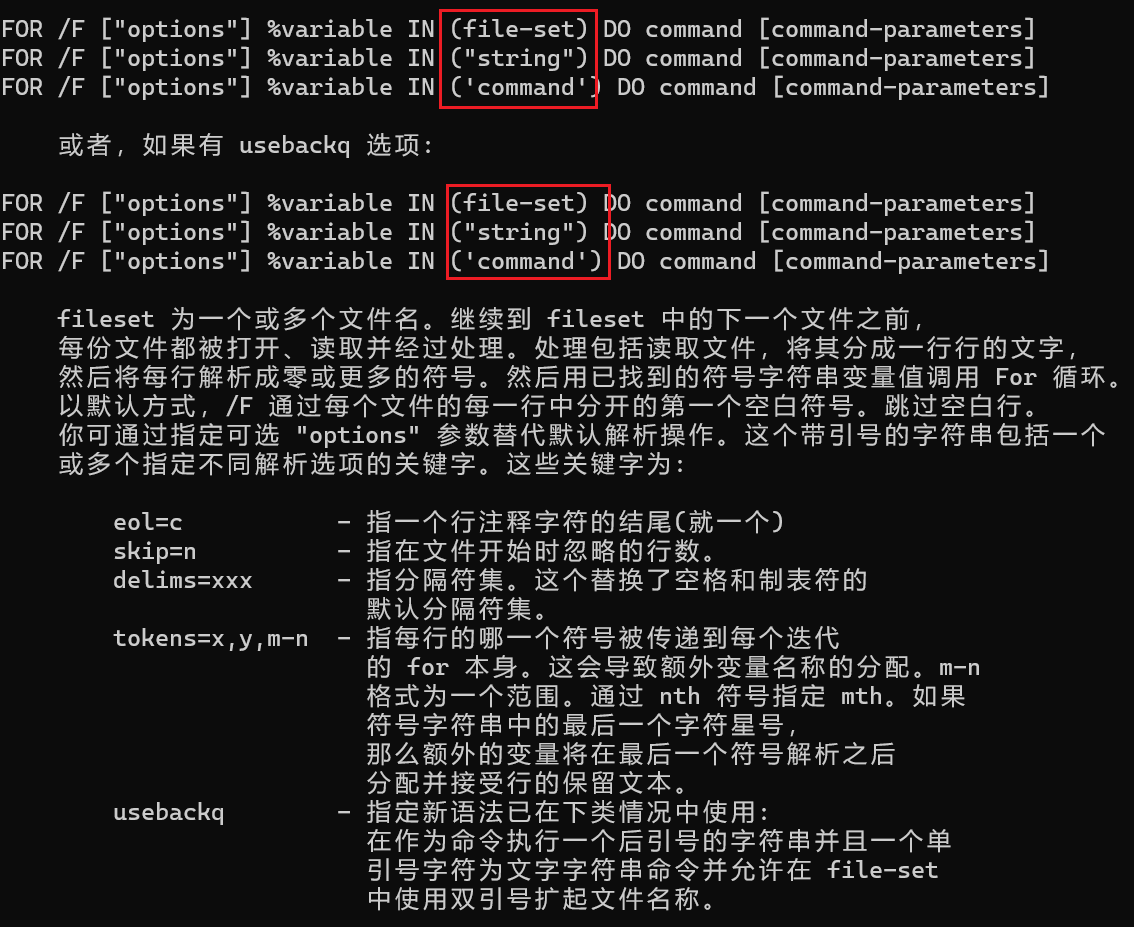
#### **/F 的常用选项**

| 选项 | 作用 | 示例 |
| --- | --- | --- |
| delims=xyz | 定义分隔符（默认空格/制表符） | delims=, 按逗号分割 |
| tokens=n | 提取第n列（多列用逗号分隔，如 1,3） | tokens=2 提取第二列 |
| skip=n | 跳过前n行 | skip=2 跳过前两行 |
| eol=; | 定义行注释符（默认 ;） | eol=# 忽略以 # 开头的行 |
| usebackq | 改变数据源引用方式（见下文） | 处理带空格路径或命令输出 |

#### **usebackq 的作用**

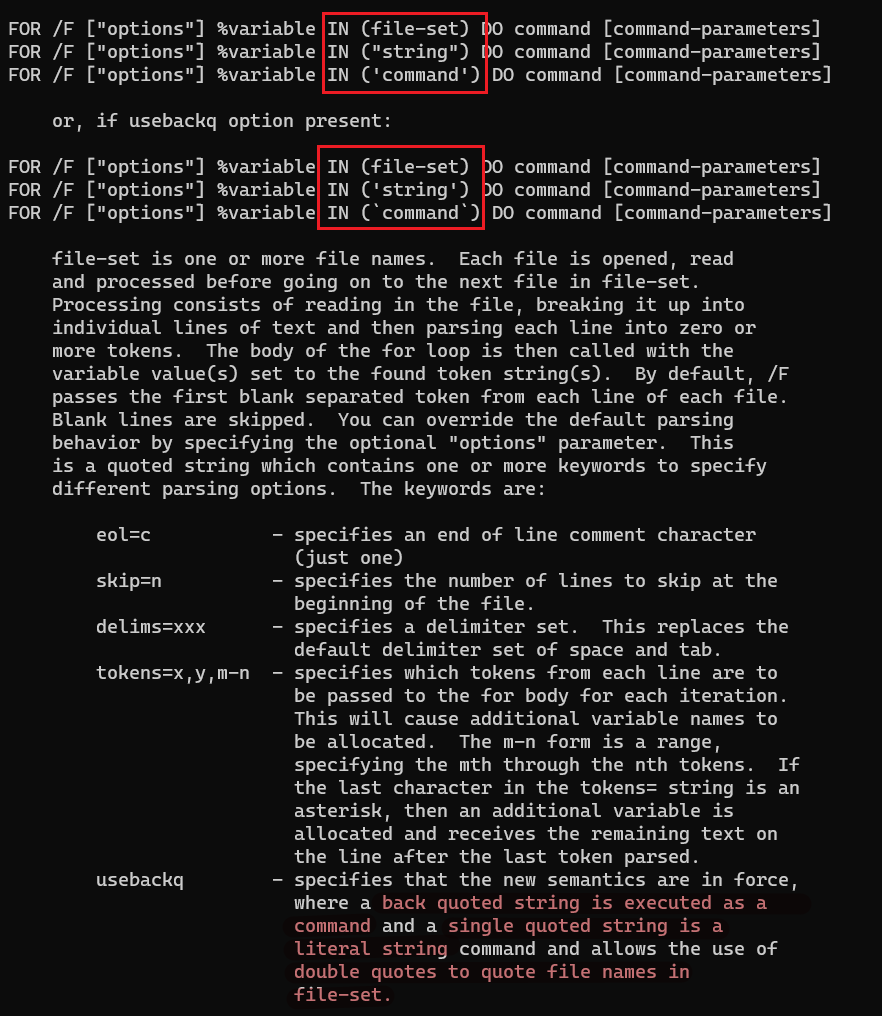
##### 此处有彩蛋：官方文档中的Bug

在windows的中文官方帮助中，对 -F 参数的说明，其实是错误的



可以看到，这里对使用 usebackq 的介绍和不使用该参数完全一样，而且，这里中文翻译的非常拗口，很难理解什么意思

不过，我们通过 chcp 65001 命令将命令行代码页切换至utf8，这时查看帮助会显示英文帮助



可以看到，理解起来就清晰多了， usebackq解决以下场景的问题：

1. **处理带空格的文件路径**：用双引号引用文件名。
2. **执行命令并解析输出**：用反引号 ` 引用系统命令。
3. **直接解析字符串**：用单引号 ' 引用字符串。

##### **案例一**：

@echo off  
for /F "usebackq delims=" %%a in ("C:\My Files\data.txt") do echo %%a

* 若不使用 usebackq，路径中的空格会导致解析错误。

##### \*\* 案例二\*\*

解析 data.csv（格式：姓名,年龄,城市），跳过标题行，提取姓名和城市。

**输入文件 data.csv**：

姓名,年龄,城市  
张三,25,北京  
李四,30,上海

批处理：

@echo off  
for /F "usebackq skip=1 delims=, tokens=1,3" %%a in ("data.csv") do (  
 echo 姓名：%%a，城市：%%b  
)

**输出**：

姓名：张三，城市：北京  
姓名：李四，城市：上海

## **应用场景总结**

| 参数 | 适用场景 | 典型命令示例 |
| --- | --- | --- |
| 无参数 | 遍历简单文件或字符串列表 | for %%a in (\*.txt) do echo %%a |
| /R | 递归处理目录下的文件 | for /R %%a in (\*.log) do del %%a |
| /D | 仅处理目录 | for /D %%a in (C:\\*) do echo %%a |
| /L | 生成数字序列（如循环固定次数） | for /L %%a in (1,1,5) do echo %%a |
| /F | 解析文本文件、命令输出或复杂字符串 | for /F "tokens=2" %%a in (data.txt) |

所以通过掌握For命令的这些参数和技巧，完全可以让它成为批处理脚本中处理文件、文本和系统任务的“瑞士军刀”。