windows的硬链接、符号链接、快捷方式

# mklink命令

mklink是windows系统下创建**符号链接和硬链接**的命令工具，它是一个很好的解决文件系统问题的工具。使用它需要管理员权限。

首先，先来介绍下mklink这个命令：

mklink 创建**符号链接、硬链接**。

MKLINK [[/D] | [/H] | [/J]] Link Target

/D **创建目录符号链接**。默认为文件符号链接。

/H **创建硬链接而非符号链接**。

/J 创建目录联接。

Link 指定新的符号链接名称。

Target 指定新链接引用的路径 (相对或绝对)。

说明也已经很详细了，我们直接看下面例子吧。

# 创建符号链接，硬链接和快捷方式的不同

下面的图片向我们展示了在windows系统中创建**符号链接，硬链接和快捷方式**有什么不同。

## 符号链接（Symbolic link）

执行命令 **mklink link\_name target\_name**

### 创建链接后的图标和快捷方式很像, 都有一个箭头的标志

### 在系统中不占用空间

### 在文件系统中不是一个单独的文件

### 在操作系统层解析（！？）

### **如果源文件被删除了，链接就没用了**

### 移除源文件不会影响符号链接

### 移除链接文件也不会影响源文件

### win10\_x64\_build10565上测试不可以右键修改图标和设置管理员运行

### 文件大小为0字节和不占用空间

### 文件属性的创建时间和修改时间都是**软链接创建和修改时的时间**

### 文件类型是.SYMLINK

### 可以在cmd下运行软链接(假如链接的是程序, 且运行命令是XXX即可)(win10\_x64\_build10565上测试通过)

## 硬链接（Hard link）

执行命令 **mklink /H link\_name target\_name**

### 在系统中占用的空间与源文件相同，但**在系统中引用的是相同的对象（不是拷贝）**

### 在操作系统层解析（！？）

### 图标和创建快捷方式的图标不同(没有快捷方式的小箭头)

### **移除源文件不会影响硬链接**

### 移除硬链接不会影响源文件

### 如果源文件被删除，它的内容依然通过硬链接存在

### 硬链接文件的任何更改都会影响到源文件

### 文件大小, 占用空间, 创建和修改时间跟原原文件一样

### 可以在cmd下运行硬连接(假如链接的是程序)

## 快捷方式（Shortcut）

**在选择的源文件上鼠标右键，通过下拉菜单创建**

### **快捷方式在系统中跟源文件是完全分离的**

### 只有那些懂得快捷方式的程序知道它们

### 如果源文件删除，链接就没用了

### 移除源文件不会移除快捷方式

### 移除快捷方式不会影响到源文件

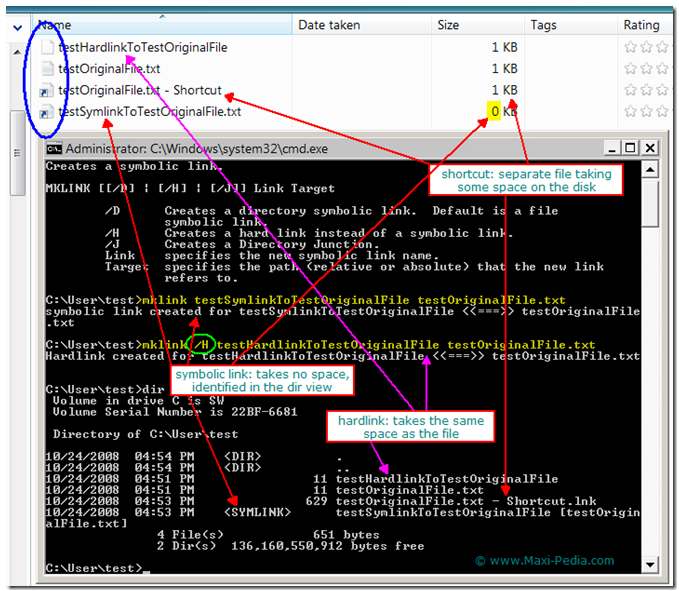
### 可以右键更改图标或者设置管理员运行

### 文件属性的创建时间和修改时间都是快捷方式创建和修改时的时间

### 文件大小仅有几百字节, 跟源文件大小无关

### 文件类型是.lnk

### 可以在cmd下运行快捷方式(假如链接的是程序, 且运行命令是XXX.LNK)(win10\_x64\_build10565上测试通过)



# mklink 硬链接和符号链接的区别

**硬链接只能用于文件，不能用于文件夹，而且硬链接和目标文件必须在同一个分区或者卷中。**硬链接的目的是为了给文件创建多个目录路径，而不像符号链接是为了指向某个已有的文件。

假设要给Target. txt文件创建一个硬链接，系统下载可以执行以下命令：

**mklink/H Link.txt Target.txt**

**和符号链接一样，硬链接中所做的任何修改，都会自动应用到目标文件上。**但是硬链接具有以下一些不同的地方。

(1)**硬链接必须引用同一个分区或者卷中的文件，而符号链接可以指向不同分区或者共享文件夹上的文件或者文件夹**。

(2) 硬链接只能引用文件，**而符号链接可以引用文件或者文件夹**。

(3)Windows会自动维护硬链接，即使把硬链接复制到其他文件夹，硬链接和目标都可以继续访问。

(4)**删除目标文件，硬链接可以继续保留**。**只有把目标文件和所有的硬链接都删除，才能把该文件彻底删除**。

(5)如果win7把符号链接的目标文件删除，然后用一个同名文件替换，则符号链接会指向新的目标文件；而把硬链接的目标文件删除’再用同名文件替换，则硬链接还是会继续引用原始文件。

(6)也就是说，硬链接和目标文件的地位相等。事实上，原始的目标文件本身也相当于硬链接，新建硬链接，只是相当于增加一个目录路後而已。

(7)**硬链接看上去和真的文件一模一样(实际上就是真实的文件)，不像符号链接那样有一个快捷方式的小箭头，但是硬链接并不会增加磁盘空间的占用**。

(8)对硬链接进行NTFS权限的修改，会同时影响到目标文件(因为两者等价)，而符号链接和目标文件可以设置不同的NTFS权限。

# mklink /D和/J的区别

**目录符号链接和目录联接**（看原文即目录的硬链接）的区别在于：

目录联接在创建时会自动引用目标目录的绝对路径，而符号链接允许相对路径的引用。

如分别用 mklink /D dira tdir 和 mklink /J dirb tdir 创建 dira、dirb 对相对目录的 tdir 的符号链接和目录联接，之后将 dira、dirb 移动到其它目录下，则访问 dira 时会提示“位置不可用”，访问 dirb 时仍然正常指向 tdir；

且win10\_x64\_build10565的cmd下dir命令查看会发现, dira符号链接(软链接)链接到的是相对路径下的tdir文件(不管是否存在tdir文件), 且文件类型是symlink, dirb目录联接则链接到绝对(全)路径下的tdir文件, 且文件类型是junction(可能是系统自动把相对路径转换为全路径)

而分别用 mklink /D dira c:\demo\tdir 和 mklink /J dirb c:\demo\tdir 创建 c:\demo\tdir 的符号链接和目录联接，再将这两个目录链接移动到其它目录下，则 dira 和 dirb 均可正常指向 c:\demo\tdir；

**由此可见当创建目录链接时对目标目录使用绝对路径，D 和 J 两个参数实现的目录链接效果是一样的；**

英文原文：

**MKLINK [[/D] | [/H] | [/J]] Link Target**

/D Creates a directory symbolic link. Default is a file symbolic link.

/H Creates a hard link instead of a symbolic link.

/J Creates a Directory Junction.

/D creates a symbolic link, or a soft link.This essentially acts like a shortcut to a folder in prior versions of Windows, except you don’t have to use an actual shortcut.

/H creates a hard link, which points directly to the file.This option can’t be used for folders directly for some reason, you’ll have to use the next option.

/J creates a “Directory Junction”A Directory Junction is actually just a hard link to a directory. This is a feature that existed prior to Vista as well. If you are trying to symlink to a directory using a hard link, then you should use this option.

Understanding Hard vs Soft Links================================Hard Link

A hard link directly points to the file, and acts to the operating system as if it is the file itself. You’ll want to use this option the majority of the time if you are trying to fake an application’s directory.

**Soft Link**

A soft link is essentially a shortcut to a file or folder – if you are using Windows explorer, you’ll be redirected to the directory if you double-click on a shortcut, it won’t pretend its part of the filesystem. You can still directly reference or open a file with the symlinked path, and it mostly works.