MKLink

Create a symbolic link to a directory or a file, or create a hard file link or directory junction.

```
Syntax
      MKLINK [[/D] | [/H] | [/J]] LinkName Target

Key:

   /D     Create a Directory symbolic link. (default is file)

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

   /J     Create a Directory Junction.

   LinkName The new symbolic link name.

   Target The path (relative or absolute) that the new link refers to.
```

Symbolic links are the newest and most flexible type of link, (first introduced in Vista) they are transparent to users; the links appear as normal NTFS files or directories, and can be acted upon by the user or application in exactly the same manner. Symbolic links can span volumes and can use UNC paths. Symbolic links are also the only type of link which can be set to a *relative* path (on the same volume).

A symlink can be an absolute path `C:\Programs` or a path relative to the link's location `\Programs`.

Types of Link

	Can link to files?	Can link to folders?	Can link across hard drives?	Can point to a non-existent target?	How to delete:
Shortcut	Yes	Yes	Yes	Yes	Del
Hard link	Yes	No	No	No	Del
Junction (soft link)	No	Yes	Y (on the same computer)	Yes	RD
Symbolic link	Yes	Yes	Yes	Yes	RD *folder* or Del *file*

Symbolic Links and Directory Junctions are implemented using reparse points.

Hard Links are implemented with multiple file table entries that point to the same inode – the same as Unix hard links. If the original filename is deleted, the hard link will still work - it points directly to the data on disk.

It is possible (but not advisable) to create two links that point to each other in a loop, or a link that targets itself. Symbolic links can expose security vulnerabilities in applications that aren't designed to handle them.

Unfortunately under Microsoft Windows neither hard links or symbolic links are supported by .zip files.

List existing Links and Junctions

The standard DIR command will display Symbolic Links, indicated with `<SYMLINKD>`
The DIR /A:S command will display Junctions, indicated with `<JUNCTION>`

```
DIR /A:S %userprofile%
```

Elevation

By default, only Administrators can create symbolic links. The security setting 'Create symbolic links' can be granted at:
`Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\`

Creating a symbolic link requires elevation, but from Windows 10 build 14972, symlinks can be created without needing to elevate the console as administrator - this does however require that you have Developer Mode enabled.

Windows Explorer - drag and drop

- Selecting a symbolic link in Windows Explorer will select the original directory.
- Selecting a junction in Windows Explorer will select the junction.

- Dragging a symbolic link to a new directory in Windows Explorer will move the symbolic link to the new directory.
- Dragging a junction to a new directory in Windows Explorer will move the original directory to the new directory.

Errorlevels

If the link was successfully created %ERRORLEVEL% = 0
Link could not be created or bad parameters given %ERRORLEVEL% = 1

Examples

Link for a file:

```
C:\> MKlink ss64.exe C:\Windows\system32\notepad.exe
C:\> Dir
C:\> Del ss64.exe
```

Link for a folder:

```
C:\> MKlink /D Apr C:\work\April
C:\> Dir
C:\> RD Apr
```

MKLINK is an internal command.

"And so its my assumption, I'm really up the junction" ~ Squeeze

Related commands

FSUTIL - Create Hard Junction Link.
FSUTIL behavior set symlinkevalution - Allow/disable symbolic links.
MOUNTVOL - Manage a volume mount point.
RD - Delete folder or Junction Point.
PowerShell: `New-Item -ItemType SymbolicLink`
Q205524 - How to create and manipulate NTFS junction points.
Hard Links and Junctions - MSDN
Junction - Create directory symbolic link (sysInternals).