Windows Symbolic and Hard Links

Introduction

Windows has had Unix-style symbolic links and hard links since NTFS 5.0 (meaning Windows 2000 and later), but many Windows users don't seem to know about them. These are **not** the relatively useless *Windows Shortcuts* that appear on your Explorer context menu when you right-click. The article Windows Symbolic and Hard Links on shell-shocked is a great article on the subject, but I thought I would put a little quick HOWTO blurb here for GLAST users.

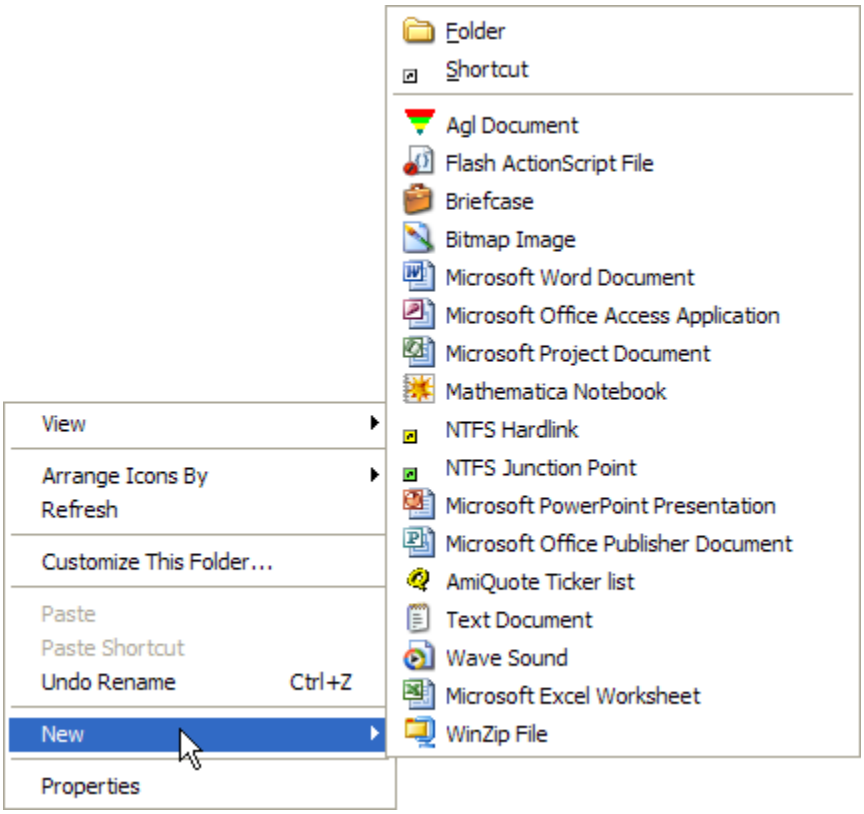
All of the following tools are free (as in price).

NTFS Link (for GUI)

Although this functionality is built-in to Windows 2000/XP, there are several extensions available to make working with symlinks more comfortable from Explorer since Microsoft didn't include this functionality in the context menu. My favorite tool is NTFS Link.

NTFS Link is a Shell Extension that provides its functionality through the Explorer context-menu when you right-click. In the figure below, notice the NTFS Junction Point and NTFS Hardlink options. Another cool feature is that the extension knows about outstanding symlinks, so that if you move, remove or rename the target, the shell extension will warn you and give you the option of updating all of the references.

Download NTFS Link (500KB)
Download NTFS Link Source (70 KB)



Junction (for command-line)

For you command-line users, there is the `junction` command by Mark Russinovich to make command-line useage more pleasant. This would probably be really useful for our Windows installer area when we get that set up.

Download Junction (16KB)
Download Junction Source (22 KB)

This is the blurb (cut-and-pasted) from the Sysinternals about the `junction` command.

Win2K's version of NTFS supports directory symbolic links, where a directory serves as a symbolic link to another directory on the computer. For example, if the directory D:\SYMLINK specified C:\WINNT\SYSTEM32 as its target, then an application accessing D:\SYMLINK\DRIVERS would in reality be accessing C:\WINNT\SYSTEM32\DRIVERS. Directory symbolic links are known as NTFS junctions in Win2K. Unfortunately, Win2K comes with no tools for creating junctions - you have to purchase the Win2K Resource Kit, which comes the linkd program for creating junctions. I therefore decided to write my own junction-creating tool: Junction. Junction not only allows you to create NTFS junctions, it allows you to see if files or directories are actually reparse points. Reparse points are the mechanism on which NTFS junctions are based, and they are used by Win2K's Remote Storage Service (RSS), as well as volume mount points.

If you want to view reparse information, the usage for Junction is the following:

Usage: junction [-s] <directory or file name>

-s Recurse subdirectories.

If you want to create or delete a junction, use Junction like this:

Usage: junction [-d] <junction directory> [<junction target>]

To delete a junction specify the -d switch and the junction name.