Link Shell Extension

Last Updated March 14th 2006, Version 2.100

Introduction The NTFS file system implemented in NT4, Windows 2000, Windows XP and Windows XP-64 supports a facility known as **hard links** (referred to herein as **HardLinks**). HardLinks provide the ability to keep a single copy of a file yet have it appear in multiple folders (directories). They can created with the POSIX command ***ln*** included *in* the Windows Resource Kit or the *fsutil* command utility included in Windows XP. Thus, using standard Windows facilities HardLinks can only be created at the command prompt, which can be tedious, especially when HardLinks to multiple files are required or when one only makes occasional use of HardLinks. Support for Junctions in standard Microsoft software offerings is even more limited than that offered for HardLinks.

Link Shell Extension (LSE) provides for the creation of HardLinks, Junctions, and Vista's Symbolic Links, (herein referred to collectively as Links) and a Folder Cloning process that utilises HardLinks or Symbolic Links. LSE, as its name implies is implemented as a Shell extension and is accessed from Windows Explorer, or similar file/folder managers. The extension allows the user to select one or many files or folders, then using the mouse, complete the creation of the required Links - HardLinks, Junctions or Symbolic Links or in the case of folders to create Clones consisting of Hard or Symbolic Links.

LSE is supported on all Windows versions that support NTFS version 5.0 or later, including Windows XP-64 and the upcoming Vista operating system. HardLinks, Junctions and Symbolic Links are NOT supported on FAT file systems, and nor is the Cloning process supported on FAT file systems.

Within this document the terms ***action button*** and ***action (pop up) menu*** are used to refer what are often referred to as the right mouse button and the pop up menu that is displayed when that mouse button is pressed (often referred to as the context menu). Recognising that people swap the usage of their mouse buttons, Microsoft refer to the *primary* and *secondary* mouse buttons. We prefer to refer the mouse buttons as the **Select** button and the **Action** button; and rather than terms such as Context Menu, Shell Menu, Right Mouse Menu we use the term **Action** menu.

Installation The current user must have administrator privileges in order to install the software. If upgrading from a version prior to version 2.0 you must uninstall that version before installing version 2.1.

I'm not sure what message you were trying to communicate in the original text of the prior paragraph, I understood as "if you have a version earlier than 2.0 already installed, then you should uninstall it before installing Version 2.1"., which is what I've tried to convey in my revised text. However a pedant might interpret your original text as meaning, "If you have a version earlier than 2.0 (e.g. Version 1.7) then you should uninstall that version, then install version 2.0, then upgrade to version 2.1", that didn't seem right to me. There are a lot of pedants in the IT industry.

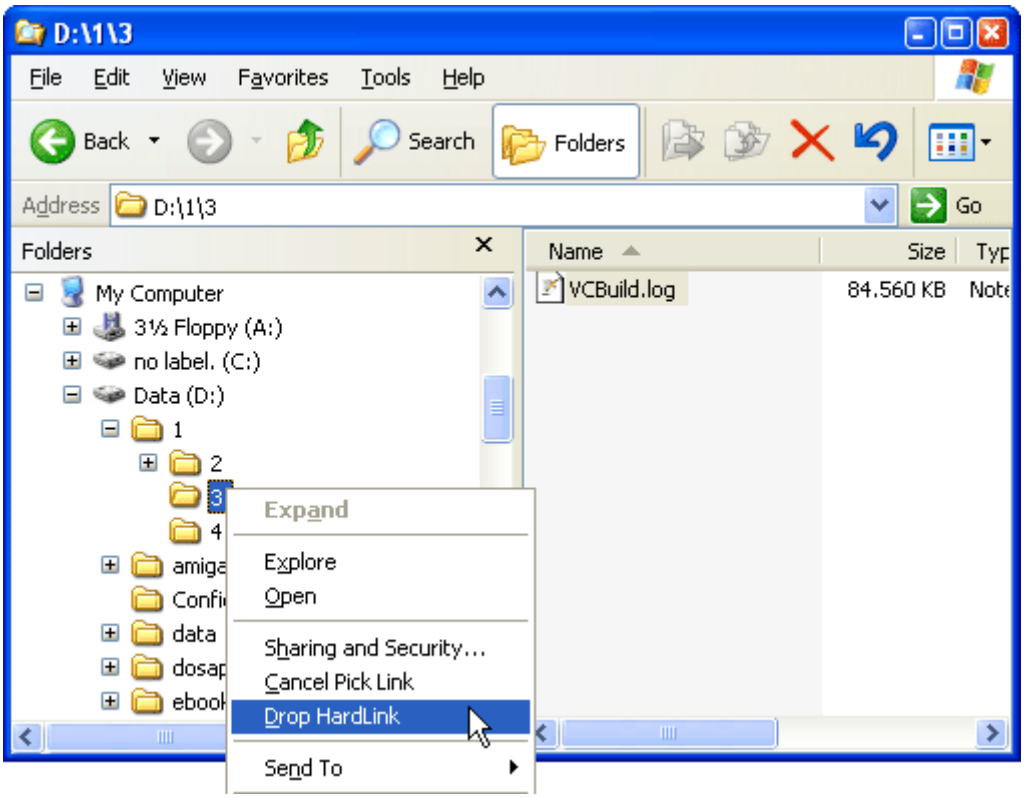
LSE is installed by executing the install program (LSESetup.exe). Follow the instructions issued by the program, there are no mandatory inputs required during installation, it is possible to change the location into which LSE is installed, the default is

C:\Program Files\LinkShellExt\

I can't recall whether the install package is bundled inside a zip archive, which is what the original text seems to imply. If it is then I would have extracted the install (setup) file from the zip archive, and then executed it to install the software. I retain software installs, but I discard the zip layer if it only consists of a single setup file and I often rename the install file to something more meaningful than “setup.exe”. The benefit gained from zipping up a one file install is trivial and IMO a waste of your time and more importantly mine. Also, this information about installation is only revealed after you've installed the software - one has to ask, why are we bothering.

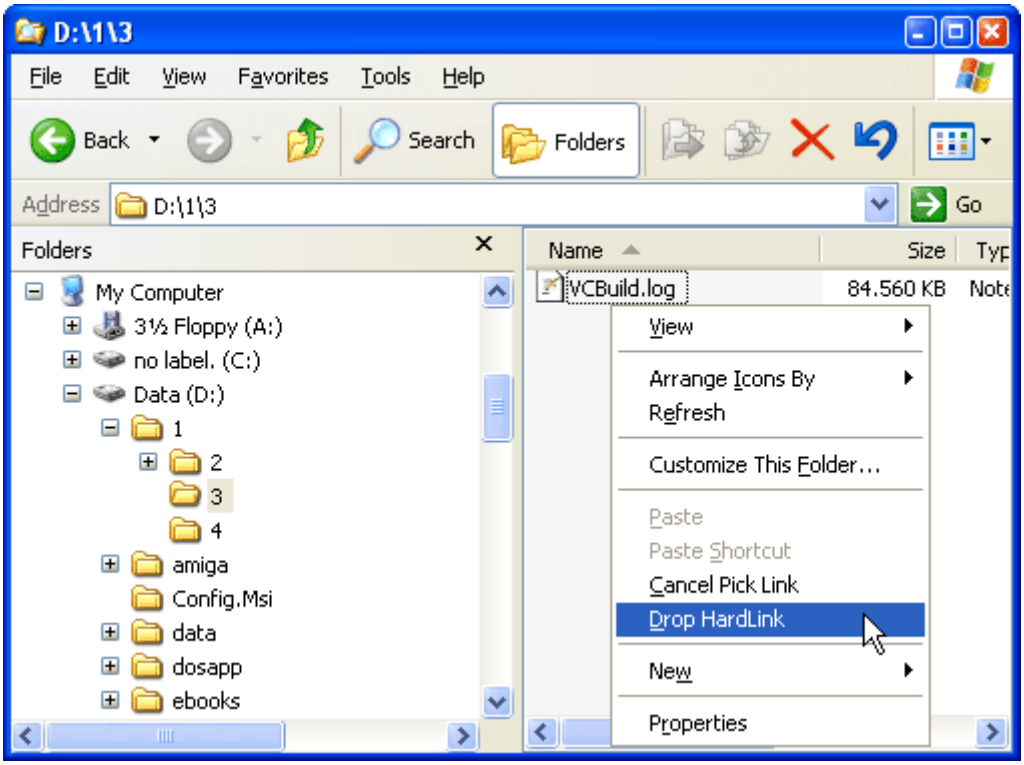
Using Link Shell Extension

Set Source Links causes the the selected files to be "stored" as the source for the HardLinks that you want to create. To create the HardLinks a destination folder must be chosen, by clicking the mouse action button on the destination folder a menu will pop up, which will include the entry - **Drop HardLink**.

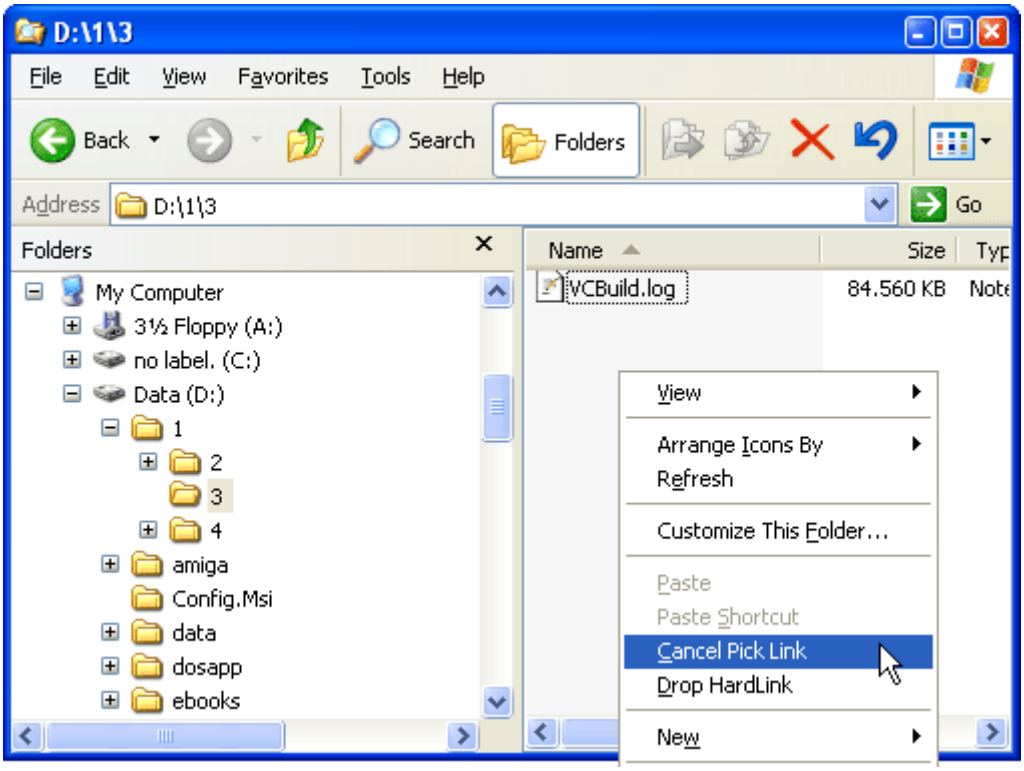


Choosing **Drop HardLink** will create the HardLinks in the selected destination folder.

In Windows 2000 and XP HardLinks can also be dropped via an Action button click in the "white space" of Windows Explorer's right pane and choosing **Drop HardLink** from the popup menu. It should be noted this feature is **only guaranteed to work in Windows Explorer**; many Explorer replacements implement an application specific white space action menu that is not readily accessible from general purpose shell extensions such as LSE.



Cancel Current Operation When the destination folder is actioned then as well as the Drop HardLink option the option of cancelling the current operation is available via the **Cancel Link Creation** entry.



Popup Submenu Since LSE supports Junction, Clones and with Vista Symbolic Links, when one or more folders are selected as the Source Links they can be dropped in several forms.

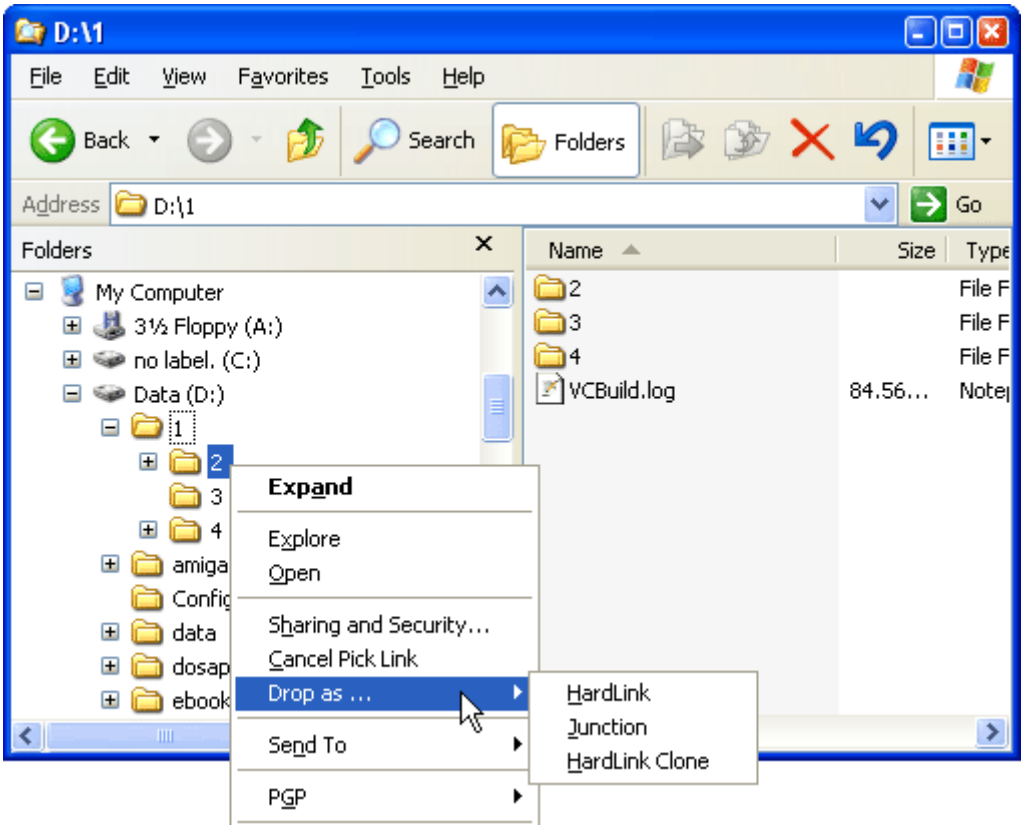
Introducing **Clones** at this point is confusing, they should (and have been) referenced in the Introduction along with HardLinks, Junctions and Vista's Symbolic Links.

To avoid crowding the popup menu, a submenu is provided that contains the different types of Links

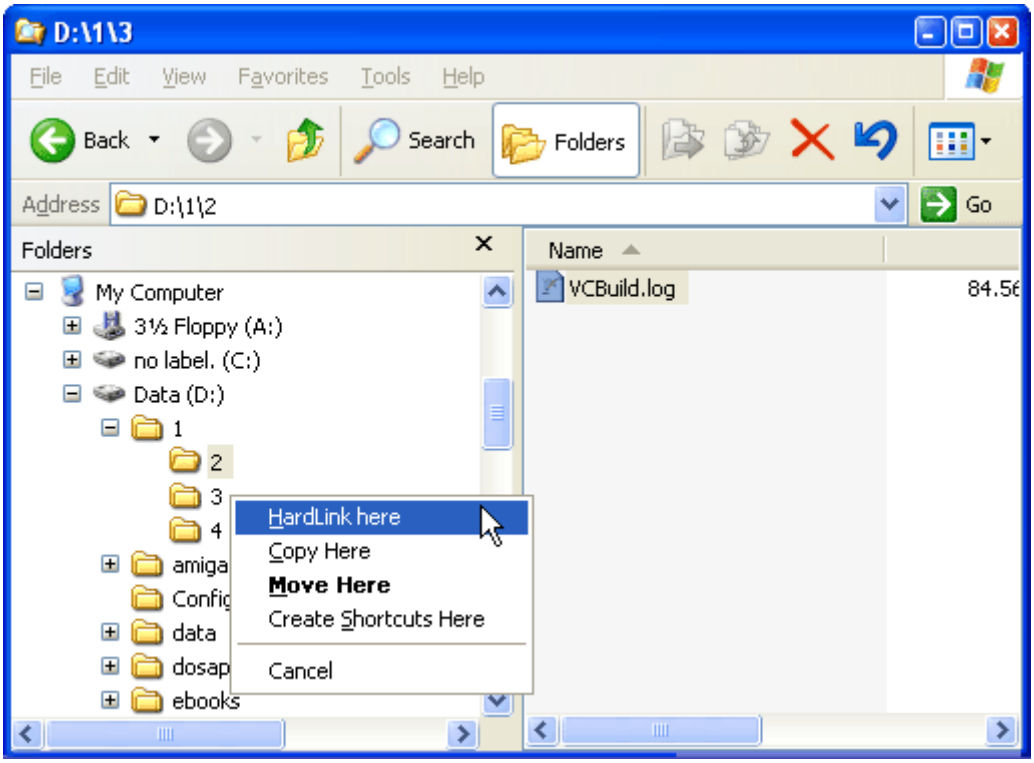
applicable to folders.

The version I have only presents a submenu when my selection is a folder, the submenu contains two choices Junction and HardLink Clone. This makes sense as Drop HardLink applies to files whereas Junctions, and HardLink Clones apply to folders,.I don't have access to Vista so I won't have much to say about Symbolic Links.

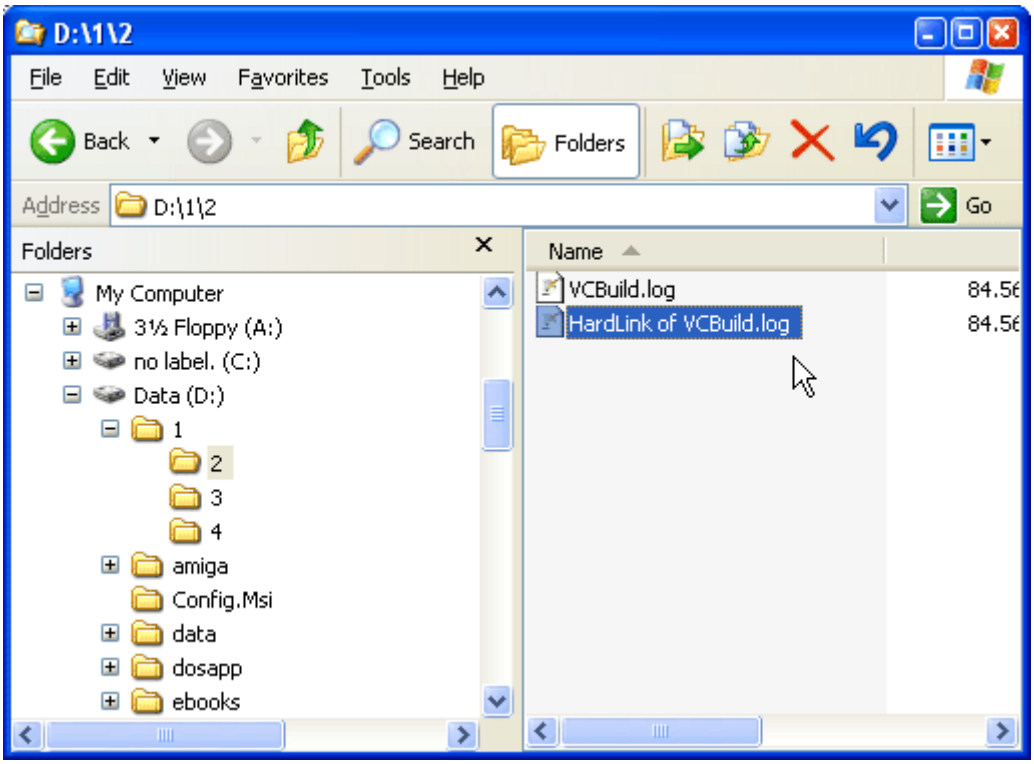
As an aside, I'm puzzled as to the difference between a Symbolic Link and a Shortcut, guess I'll find out in due course (assuming I live long enough, or should that be longhorn enough).



Drag and Drop Support Creating HardLinks via drag and drop is supported, after selecting one or more files you can drag them to the destination folder with the Action button held down; when it is released choose **HardLink Here** from the action menu to create the HardLinks of the selected files in the destination folder.



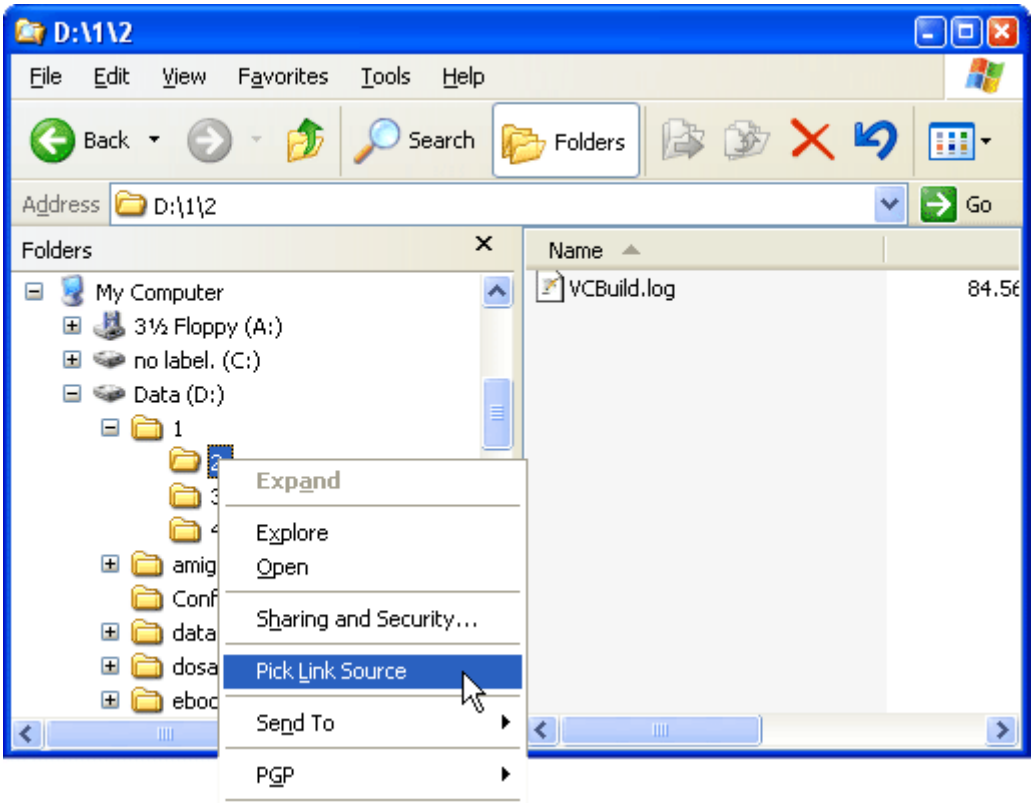
Auto Rename Files can be hard linked to the same folder as the source folder. Because two directory entries cannot have the same name, LSE uses 'HardLink of $filename' as the name of the the new link.



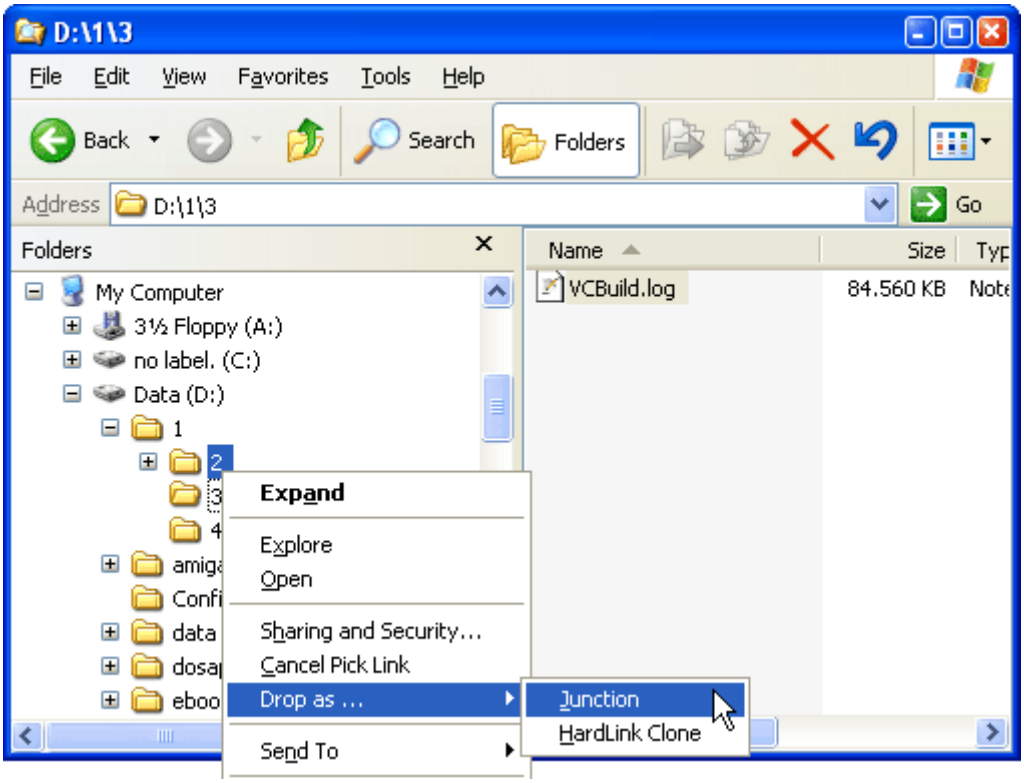
There is an inconsistency here, why should I be allowed to create a “duplicate” link, in the source folder and nowhere else – see Suggestions document for further discussion.

Junction Support Junctions are a feature of NTFS version 5.0, they provide for the creation of linkages among directories, Junction were not supported in NTFS Version 4.0

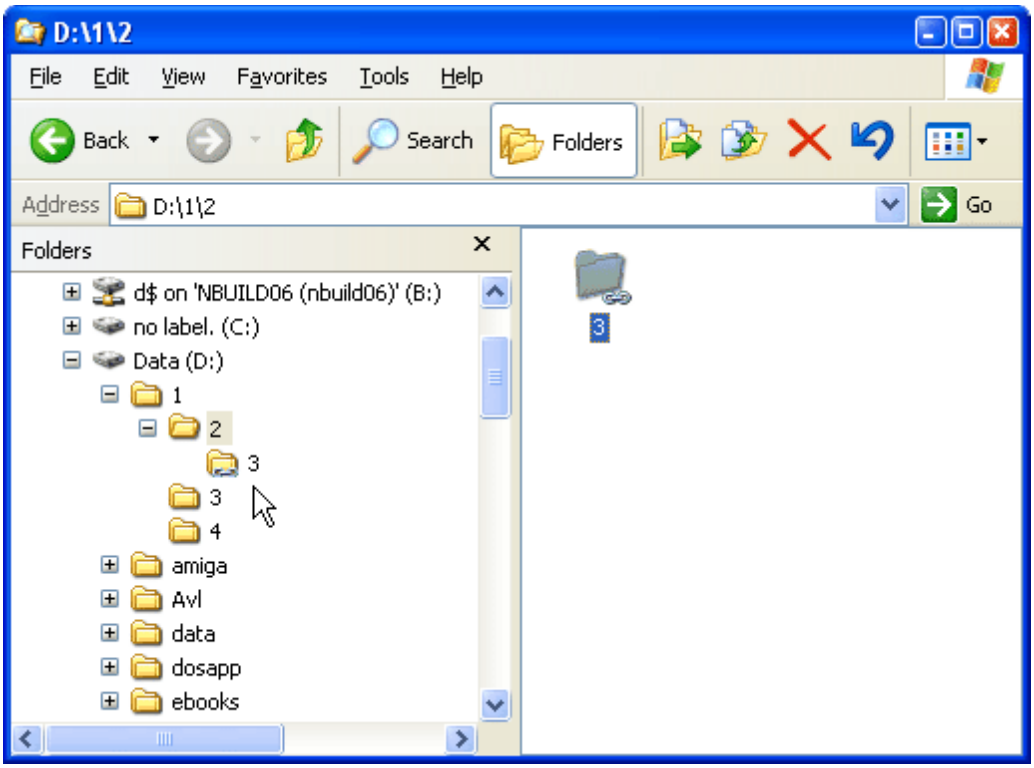
I believe it was possible to use NTFS Version 5.0 under Windows NT4.1, so strictly speaking Junctions are an NTFS version issue not a Windows NT version issue. as is implied in your original text.



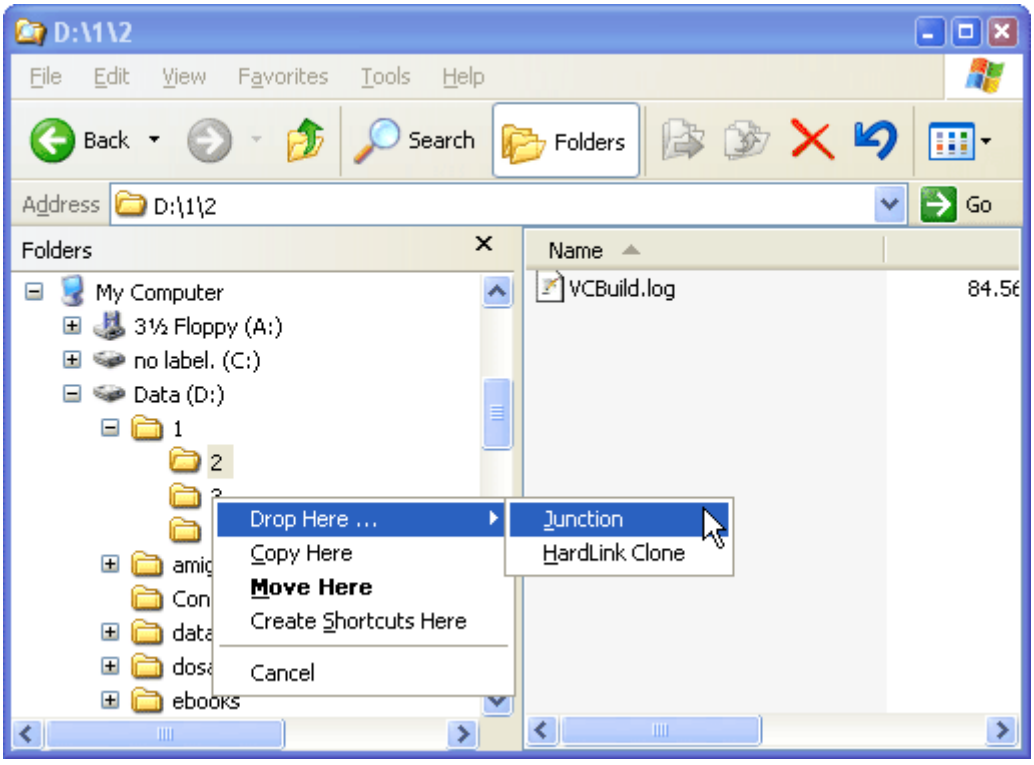
Junctions are created in the same way as HardLinks, except that the Source Link is a folder rather than a file. Select a folder, click the right mouse button, choose **Set Source Link** from the action menu, navigate to the destination folder, click the action button, open the submenu **Drop As ...** and select **Junction**:



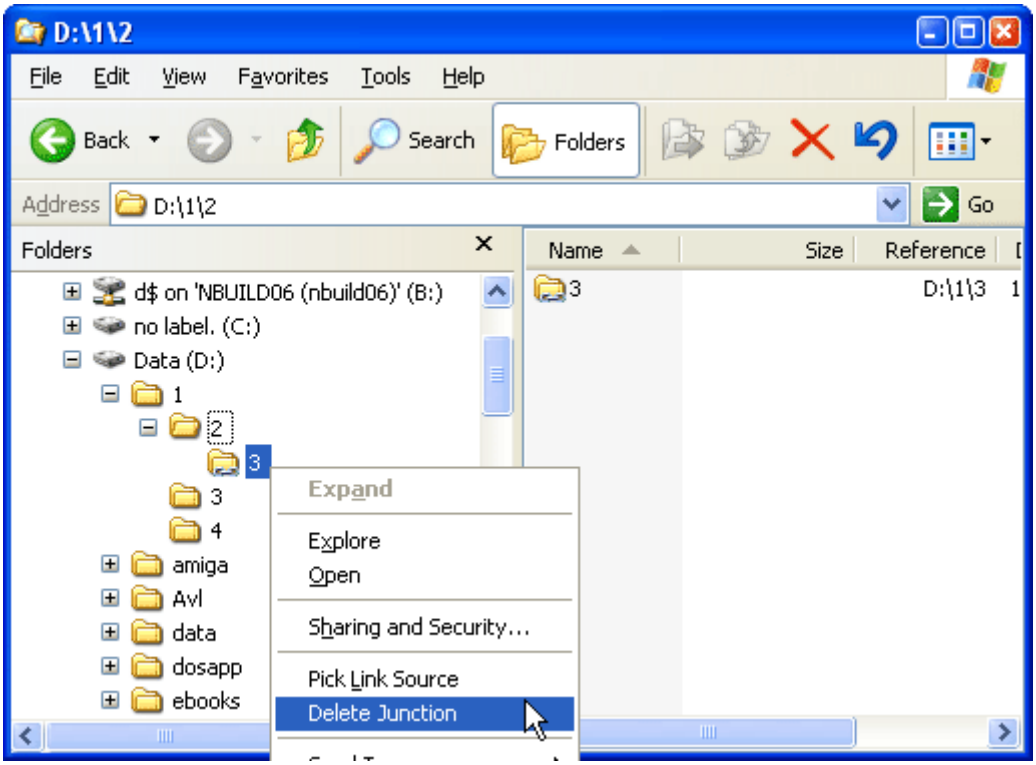
Junctions are marked with a small piece of chain below the folder icon.



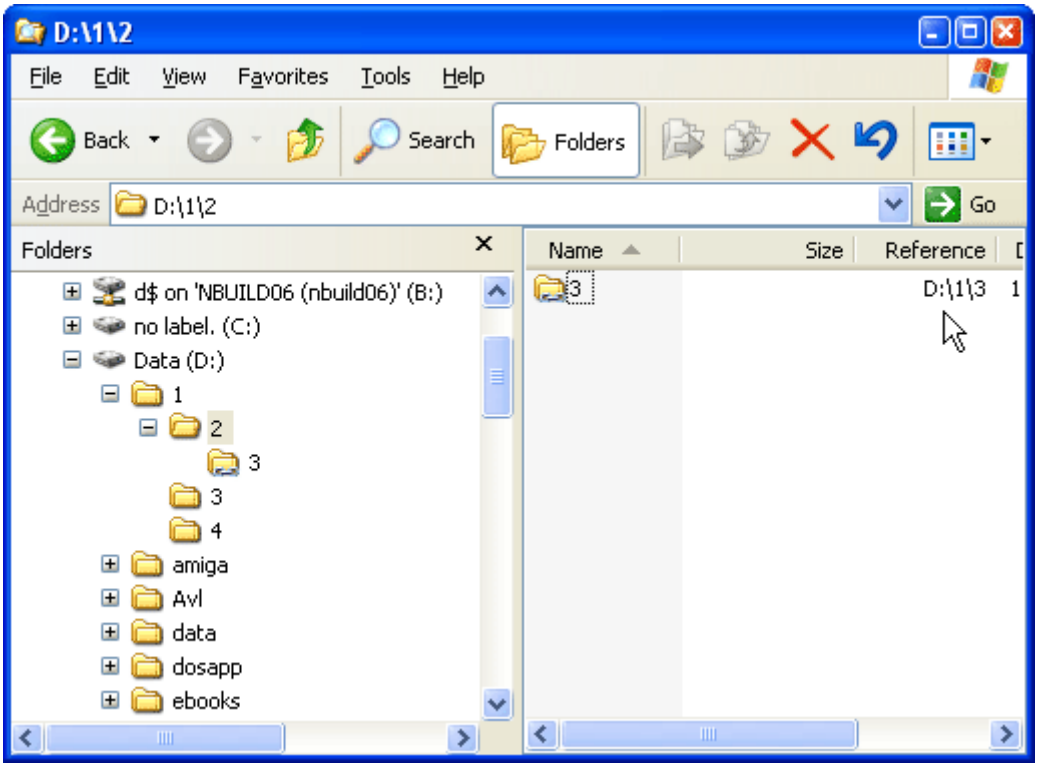
Junctions can also be created via Drag and Drop when the selected folders are dragged with the action button pressed to a destination folder; when the right mouse button is released, select the **Drop Here ...** submenu and then **Junction.**



Junctions can be deleted by choosing the **Delete Junction** entry from the right click popup menu, when a Junction is selected:



To show the origin of a junction, the reference column of a junction shows the path to which the selected junction links.



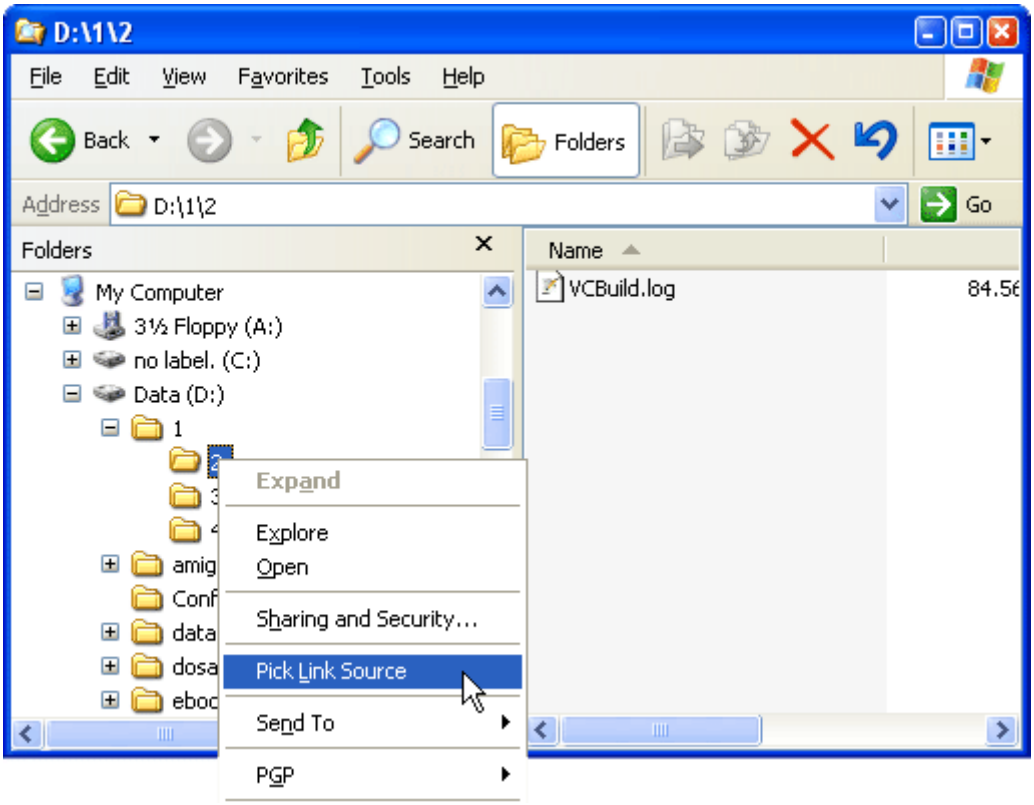
Junctions are powerful feature, but can also also be a dangerous feature. Explorer becomes unstable if you use normal delete or rename functions on junction folders. Windows Explorer will always terminate abruptly and depending on system settings, Windows XP may do likewise.

To help distinguish junction folders from a normal folders an icon overlay is implemented on junctions that shows a small two link chain under the folder icon.

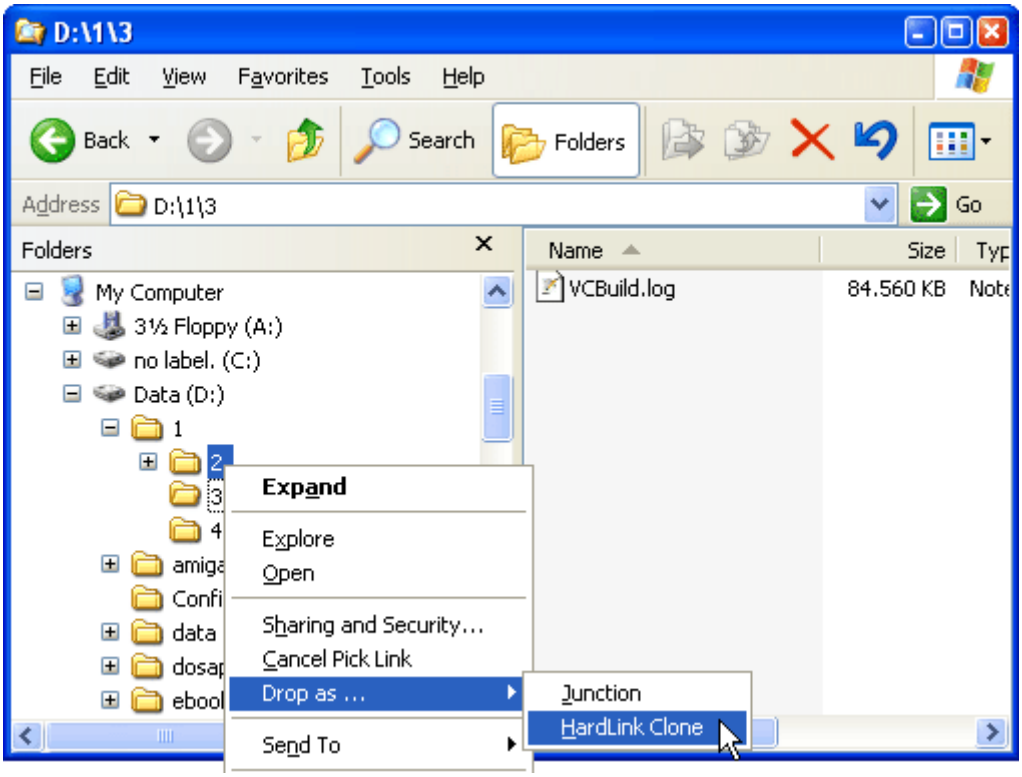
"Explorer does not delete the junction, but also the files which are in the referenced folder."

Please read my comments in the Suggestions document – in my experience the above statement is a myth, at least as far as XP is concerned. That is not to say that the normal delete works on Junctions, it definitely does not, but in my experience what happens is as I've described, but I've never lost any data as a result of accidentally hitting the delete key on a Junction I've just crashed Explorer or Windows XP. I have about 200 Junctions on my system.

HardLink Clones Clones are copies of a folder tree from a source location recreated at destination, however the files within the new folder tree are HardLinks to the same files that exist in the source folder tree.

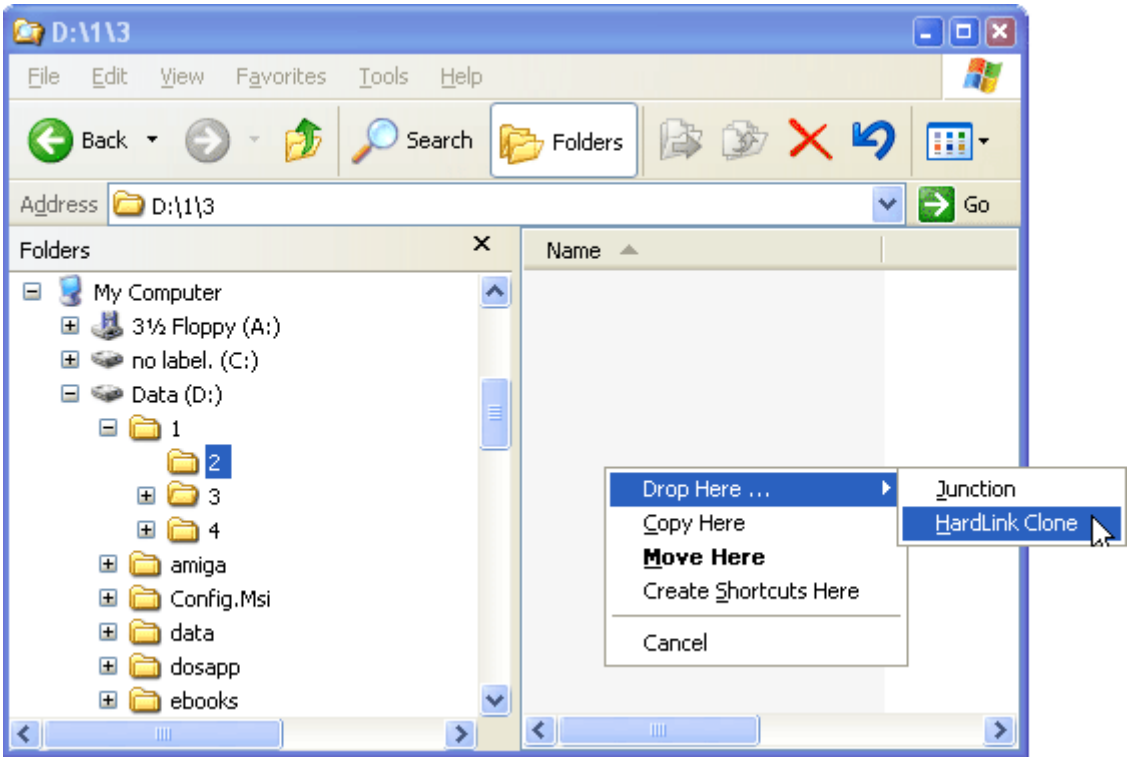


HardLink Clones are created in the same way as Junctions, select a folder, click the Action button, choose **Set Source Links** from the action menu, navigate to the destination folder, press the action button, open the **Drop As ...** submenu and select **HardLink Clone**:



HardLink Clones can also be created via Drag and Drop, select a folder and drag with the action

button depressed to a destination folder. When the action button is released open the **Drop Here ...** submenu and select **HardLink Clone**:



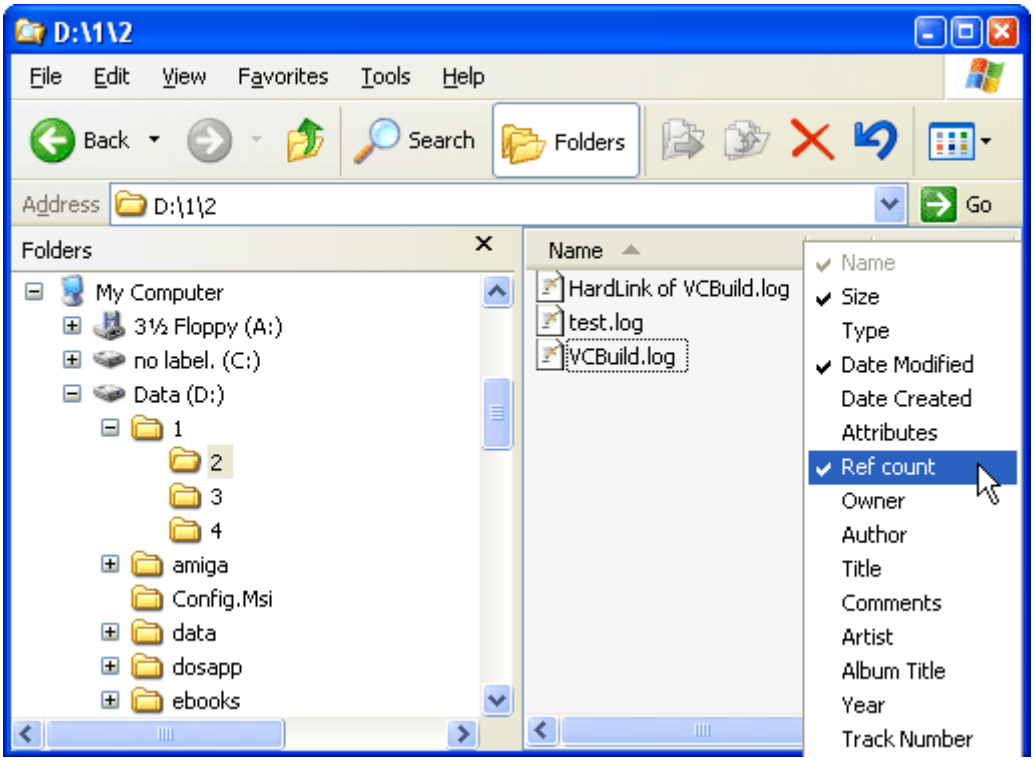
HardLink Clones are useful if you need to replicate a folder tree at a different location. The space required is minimal because the new structure consists entirely of NTFS directory entries with no actual data storage.

If both files and and folders are selected as Source Links and dropped as a **HardLink Clone** then the selected files are dropped as HardLinks alongside the HardLink Clones.

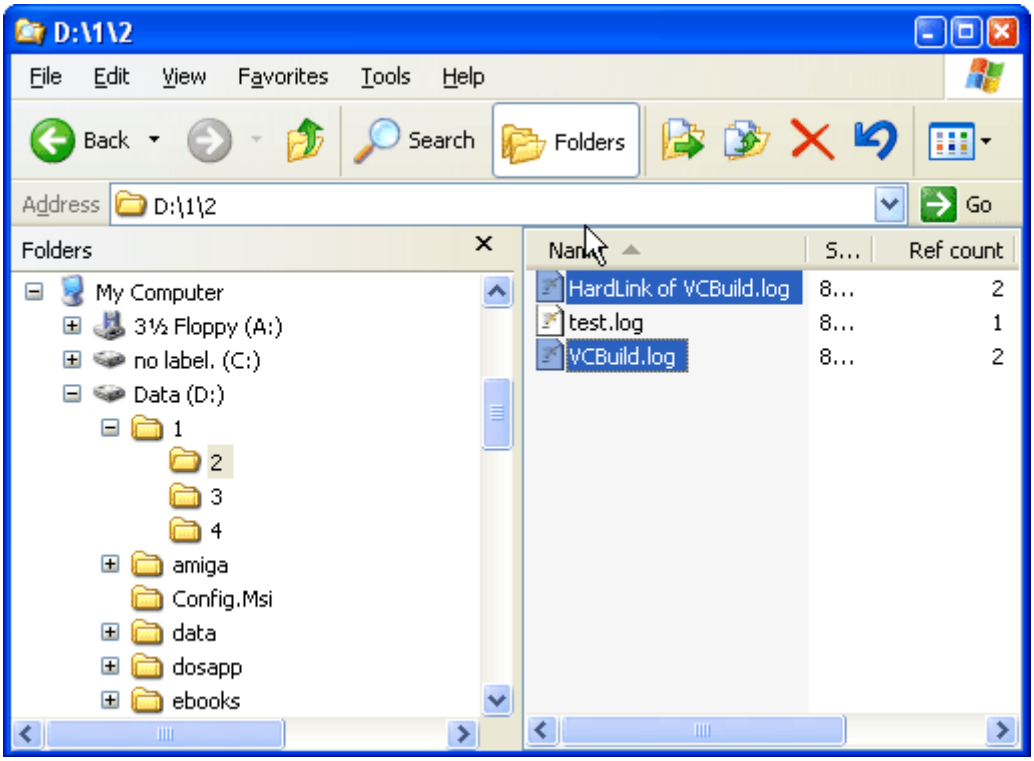
Because Clones use HardLinks they are only available within an NTFS volume. You cannot use Cloning to replicate the folder structure on one disk volume to a different volume using Hardlink Clones, because HardLinks are limited to operating on a single volume

Reference Count As described in the backgrounders section within the “default” data stream of all data objects NTFS maintains a reference count how many NTFS directory entries refer to object. In most scenarios each entry will refer a different folder, although it is possible to have multiple links to the same data object in the one folder, providing they have folder unique names

To show the reference counts, a column can be enabled in Explorers right pane by action clicking the Titles row of the details view.



After enabling the reference column the reference count is shown for each file. This feature is only available in Windows 2000 and XP, it is not supported in Windows NT4.

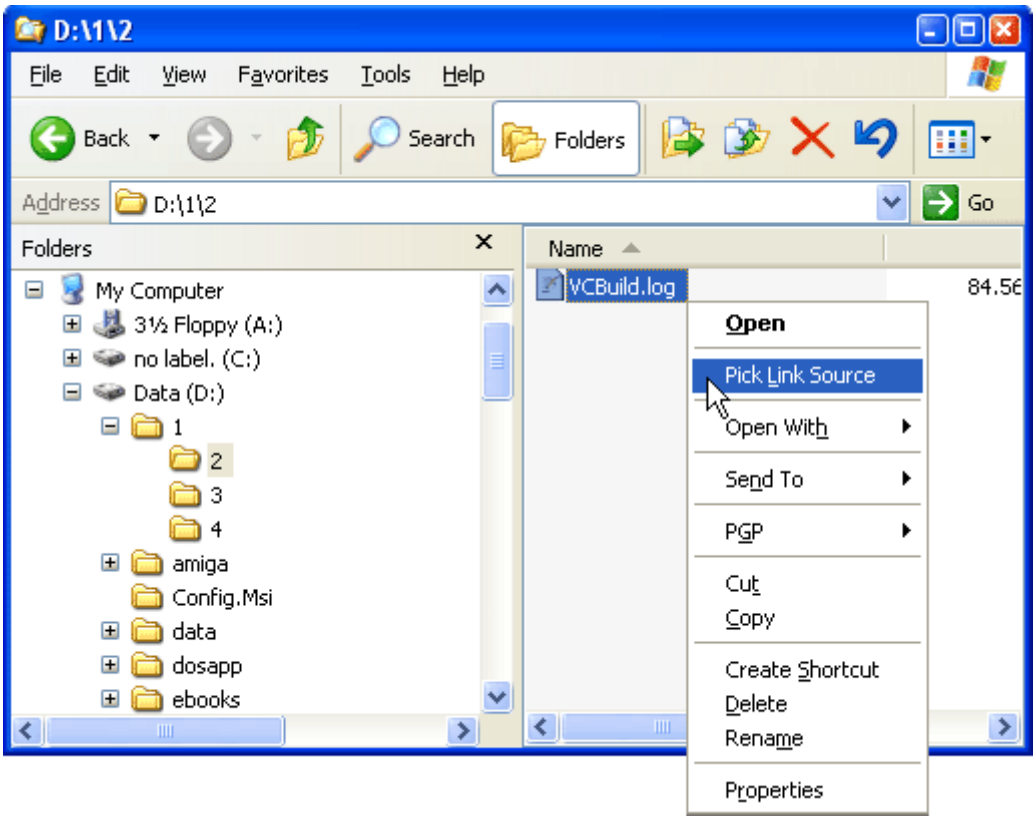


Symbolic Link & Vista

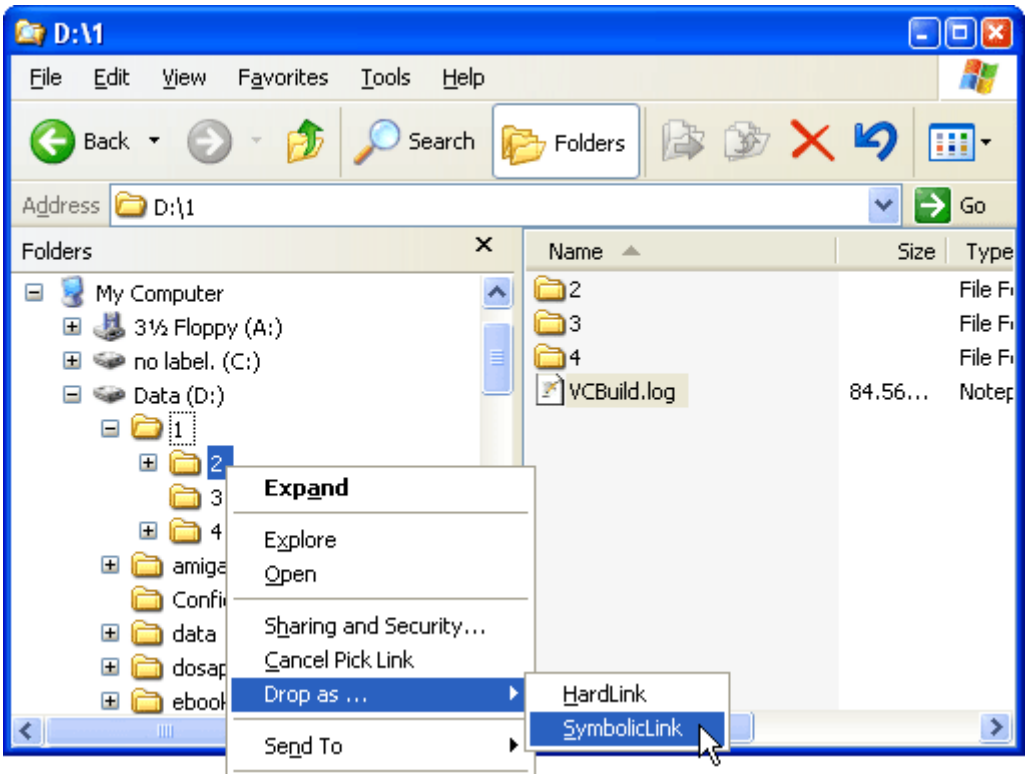
In the upcoming Vista operating system, NTFS introduces a new type of link, the *Symbolic Link.* LSE has been extended to support this type of Link.

Creating a **Symbolic Link** is essentially the same as the other Link creation processes. Action click

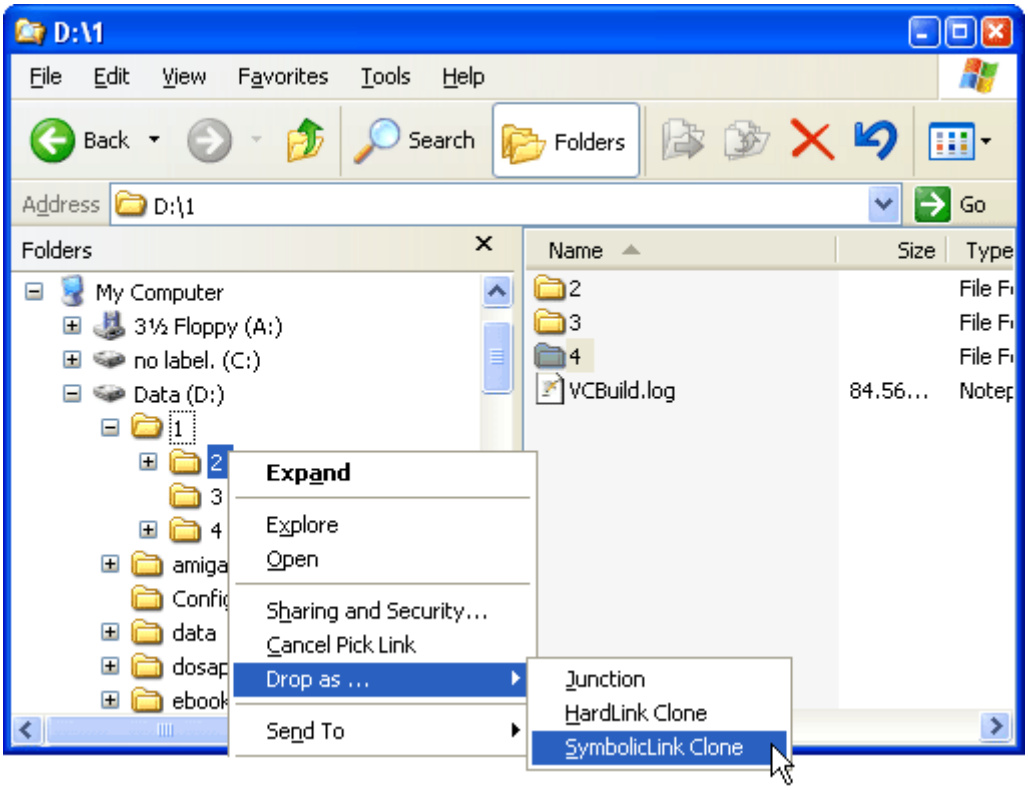
on the selected file(s) and select Set Source Link(s) from the action menu.



Under Vista when the destination folder is action clicked the menu contains a **Drop As ...** submenu, to create a Symbolic Link select SymbolicLink from the submenu. Unlike HardLinks Symbolic Links can span storage volumes.



Symbolic Links can also be selected when creating clones of existing tree structures



If both files and folders are selected as the Source Links and dropped as a **Symbolic Link Clone** then the selected files are dropped as Symbolic Links alongside the folder tree.

Backgrounders **HardLinks** are a feature common to many Unix based systems, but are not directly available with NT4/W2K/WXP. It is a feature, which must be supported by the file system of the operating system.

The above statement is untrue, there are file systems within the Unix domain that do not support HardLinks, and in all instances it is the **file system** that provides the support for HardLinks, under both Unix and Windows variants.

So what are HardLinks? It is common to think of a file as being an association between a *file name* and a *data object*. Using Windows Explorer, the file system can be readily browsed, showing a a 1:1 relationship between the *file name* and the *data object*, but this 1:1 relationship does not hold for all file systems.

Some file systems, including UFS, XFS, and NTFS have a N:1 relationship between *file name* and the *data object*, hence there can be more than one directory entry for a file.

So, how does one create multiple entries for the same data object? In Unix there is a command line utility *ln*, which is used to create link entries for existing files, hence there are many file names, or so called HardLinks, for the one data object.

For each HardLink created, the file system increments a reference count stored with the *data object*, i.e. it stores how many *file names* refer to the data object, this counter is maintained (by the file system) within the data object itself. When a file name referencing a *data object is* deleted, the *data object's* reference count is decremented by one. The *data object* itself **only** gets deleted when the reference count is decremented to zero.

The reference count is the only way of determining whether there are multiple *file name* references to a *data object*, and it only informs of their number NOT there whereabouts.

Junctions are wormholes in the tree structure of a directed graph. By browsing a Junction a maybe far distant location in the file system is made available. Modifying, Creating,.Renaming and Deleting files within a junction tree structure operates at the junction target, i.e. if you delete a file in a Junction it is deleted at the original location.

So no reference counts here, but accessing files via Junction is really working on it. Didn't understand what point you were trying to make here. To me Junctions are simply a more intelligent way of doing what people have been doing for decades with Virtual Drives.

And I thought I was the only person who thought of Junctions as wormholes, betcha you did too!

I think you need to make it clear that HardLinks do not operate across volumes, whereas Junctions do, I assume Vista SymLinks also operate across volumes, else they ain't too symbolic

Need to explain difference between SymLinks and HardLinks better, especially SymLink Clones versus HardLink Clones – I'm assuming that because one is absolute (HardLinks) and is thus constrained to operating on a single volume the other (SymLinks) is symbolic and is thus NOT constrained to operating on a single volume. Hence it should be possible to Clone a tree from one volume onto another using SymLinks rather than HardLinks.

Symbolic Links are to files what Junctions are to folders in that they are both Transparent and Symbolic. Transparency means that an application can access them just as they would any other file, Symbolism means that the data objects can reside on any available volume, i.e. they are not limited to a single volume like HardLinks. Symbolic Links differ from Shortcuts in that they offer a Transparent pathway to the desired data object, with a shortcut (.lnk), something has to read and interpret the content of the shortcut file and then open the file that it references (i.e. it is a two step process). When an application uses a symlink it gains immediate access to the data object referenced by the symlink (i.e. it is a one step process).

Limitations

- Supported platforms are NT4/W2K/WXP/WXP64.
- The Link Shell Extension can only be used with the supported platforms.
- HardLinks can only be made on NTFS volumes, under the supported platforms.
- HardLinks can only be made within one NTFS volumes, and cannot span across NTFS volumes.
- HardLinks can only be made on *fixed* NTFS volumes, you need to define "fixed", I have at least 25, 000 hardlinks on a portable 40G USB drive.
- HardLinks can only be made on local NTFS volumes.
- Junctions can only be made on NTFS volumes, under W2K/WXP.
- Junctions can only be made within one NTFS volumes, and can not span across NTFS volumes – I question this - I've got junctions on my C: drive referring to folders on my F: drive – aren't these different volumes? I've also got junctions on my portable drive that refer to folders on my "fixed" drives – I have to be careful not to to go near them when the portable drive is attached to a "foreign" machine.
- The *Set Source Link(s)* and *Drop ...* choices are only visible, if it's possible to create HardLinks/Junctions/Symbolic Links. E.G.: If you select a file on a FAT drive and press the action button, you wont see the *Set Source Links* in the action menu, because FAT file systems, don't support HardLinks/Junctions/Symbolic Links. This also happens, if you select source files on a network drive, or select a file as destination, etc. This is not quite so, see Suggestions document for at least one instance where an invalid choice is presented, although if selected it results in "no harm done"

History

Match 12th 2006 — Version 2.1 released

- Added overlay icons for junctions, so that junction visually pop into your eye.

February 27th 2006 — Version 2.0 released

- Revamped the internal structure of ShellExt.
- Introduced creation of HardLink Clones.
- Introduced submenu in the popup menu, when more than one entries would be added to popup menu to show the many dropping choices
- Support for Symbolic Links with 'Vista'.
- Support for Symbolic Link Clones with 'Vista'.
- Fixed crashes when dragging files, and using 'HardLink here'.
- Fixed problem when showing up wrong menu, when having folders disabled in the left explorer pane.
- Junctions display their origin in the reference column.
- A *Pick Link* operation can be canceled now.
- The installer restarts explorer.exe to properly add/remove the .dll
- Added an entry to Startmenu
- Support for WindowsXP64.
- Submenu for many dropping choices in popup menu

November 26th 2005 — Version 1.7 released

- Added the Delete Junction popup menu, when right mouse button is pressed on a junction.
- Fixed a handle leak in CreateJunction.

January 23rd 2002 — Version 1.6 released

- Added a Columnhandler, so that the reference count of a hardlinked file is shown in explorer. This feature only works with W2K/WXP.
- Deployment revamped so that the docu is now in .html.

October 27th 2001 — Version 1.5 released

- Revamped internal string handling to Unicode.
- Added junction support. Junctions are a feature of NTFS5, which allows to hardlink two directories.
- Added a directory background handler. This means, that after picking a hardlink it is possible to press the right mouse button on the right explorer pane and drop the hardlinks/junctions/symbolic-links.

March 23rd 2001 — Version 1.201 released

- Fixed occurence of 'Hardlink Here' if shortcuts are selected.

March 23rd 2001 — Version 1.20 released

- Added Drag and drop support

March 20th 2001 — Version 1.10 released

- Fixed the problem, that the help text was not displayed properly
- Changed the installer to the lean and mean nullsoft installer.
- Fixed the problem, that readonly files can not be hardlinked
- Fixed the problem, that hardlinks in the root dir didn't work
- Tested on W2K and HardlinkShellExt is W2K compliant

May 8th 1999 — Version 1.00 released

Status The 2.1 version is a stable version.

Acknowledgements I wish to thank those who have contributed significantly to the development of Link Shell Extension. Those include:.

Felix Kasza for the hardlink basics with NT4.
Visual C++ Developer Journal shell extension examples.
Nullsoft for the great lean and mean nsis installer
Jose Flores for the dbgtrap debug monitor.
Jean-Pierre Bergamin for the drag and drop support samples.
Travis Illig suggested to add the overlay icons for junctions, which he uses in his Junction Shell Extension.
Mark Russinovich for tipps on junction

Open Issues

- Keyboard support, eg CTRL 'a key' to drop hardlinks.
- UI compliant Snake-Progressbar for operations taking longer.
- Hardlinking .lnk files is still a bit intransparent, because of explorer.exe.
- Test coverage for Vista is low, so there might be a 2.1 covering fixes for Vista

Contact Bug reports, donations ;-) or feature requests send to Hermann Schinagl.
Download the latest version from my homepage.

Download Link Shell Extension for x86 (600k)

Download Link Shell Extension for x64 (600k)