

Preamble

Whenever I find useful free software I try to make a contribution by means other than money. I don't have much money so any financial contribution I could make would be paltry. Hence I show my appreciation by writing a detailed appraisal of the software, which is what this document is.

In the case of LSE I've also done some work on the Documentation included in the install package. I'm guessing that English is not the first language of the developers, so I thought I could make a concrete contribution in this respect. I hope you will find the changes I've made useful, there are notes in the revised document, which are highlighted **thus**.

I have also included a version of the document in PDF format, personally I prefer to have documentation of this kind as a PDF, I don't like local software that insists on having a browser available, especially to view data that is local

There are some significant inconsistencies in the documentation and elsewhere that I think need attention.

1. What is the name of the product, is it "HardLink ...", or "Link ..." or "NTFS ...", where ... can be be "ShellExtension", "Shellexension" or "Shell Extension".

I suggest that HardLink not be used, because it implies the product only deals with hard links. I also suggest that NTFS not be used because Microsoft might change the name of their file system, at one time they were using "WinFS" as the tag for their next generation file system. Which leaves us with "Link ...", which is the most generic and the least likely to suffer the vagaries of Microsoft's marketing department.

Joining words together a'la ShellExtension is a "programming convention" Whilst your product might be used by programmers, surely it is aimed at a much wider audience; people such as myself who have a lot of data stored as files that they want to manage. I'm an engineer and a writer if you're interested. Joined up words lead people to conclude that the product is a programmers tool (i.e. an API or Class Library), which it's not. So I suggest "Shell Extension".

Which gives us with "Link Shell Extension", which abbreviates to LSE, which is the same as the "London School of Economics". I think Marx went there, Keynes lectured there from time to time, and Hayek (the Austrian who gave us neo-liberal economics, and is one of Margaret Thatcher's heroes) was an LSE man.

2. The second inconsistency is the use of the words "folder" and "directory". In the context of LSE, I think only one is needed. Even though I'd prefer "directory", I recognise that "folder" has become ubiquitous. I doubt there are many people who would not know what a folder is, but I suspect there are many who don't know what a directory is – unless it's a telephone directory.
3. Thirdly there are the terms "context menu" and "popup menu", there is no reason I can see to use both, which what you do. The fact is that I don't like either of them, both are almost devoid of any meaningful information, however I acknowledge that they are the terms most often used to name the "thing" we're talking about.

My preferred term is "action menu", because that is what they are, one pops them up then in order to take an action, e.g look at a file's properties, delete a folder, edit an image with "paint", or drop a couple of HardLinks.

As a companion term I prefer to call the buttons on the mouse the Select button and Action button, it's what they're used for, to select things and action them. It also relates the mouse button and the result of clicking it together i.e. "Clicking the action button at the destination causes the action menu to pop up ...". This is surely much better than "Clicking the right mouse button at the destination will cause context menu to be displayed ...". Firstly it does not discriminate against people who swap their mouse buttons, secondly it's shorter, thirdly it conveys a message that can be understood by a greater number of people, and lastly it's heaps better than Microsoft's primary and secondary mouse button labels (which discriminates against 50% of the mouse button population). And finally one can write something meaningful on the buttons of one's mouse, 'S' & 'A' or 'A' & 'S' if they're swapped! I do some volunteer teaching at a primary school and that's what I have the children do, in water colour of course.

NTFS Link v LSE

Until now I've been using NTFS Link (NL) which I hope to replace with LSE, but there's an important feature that NL has, and that LSE doesn't. It implements icon overlays (glyphs) for hardlinks and junction folders, by default it uses a black on green "shortcut" arrow for junctions and a black on yellow one for hardlinks.

NL allows these glyphs to be changed, which I have done in the case of HardLinks, I find a black on yellow "hardlink" arrow practically indistinguishable from a black on white "shortcut" arrow, so I use my own multicolour glyph. I've included some bitmap files in the archive package, they are these glyphs and a bitmap of an icon that I use for Junction folders, you should be able to relate them to this document by their names. I tried inserting them into the text but they are "too" big and mess up the text, and I can't be bothered editing them down to a smaller canvas size.

It may help you understand my "issues" regarding things like icons, if I tell you that I use a 19" monitor running at a resolution of 1600 * 1200, and my default "View" is detail.

I'm running NL and HLSE alongside one another, when I action drag a file or files to a folder, I'm offered LSE's "HardLink Here" and NL's "Create HardLink Here", that's cool, I don't like it, but I do understand why it is so .

In this context I don't really care about having to run both NL and LSE, which ever I use works, and all the links end up with my multicolour glyph. It's sort of annoying that I have to run both for the sake of such a small, but for me, invaluable item. I'll discuss Junctions, their iconic representation, and the interaction of NL and LSE with respect thereto, subsequently.

White Space Drops

One of the features of LSE that appeals to me is the ability to select file(s) and not have to immediately action drag it/them to the target location. I use a 3 pane tabbed file browser (xplorer²), I often find that the target folder is not currently visible or may not even exist, the select and drop later feature overcomes this problem. However this feature **ONLY fully** works with Windows Explorer, it does not work with xplorer², nor with Free Commander, nor with ExplorerXP, because they each implement their own "white space" Action menus.

If after Picking the Link Source's, one Action clicks a folder then the Drop and Cancel Link options pop up. However one cannot drop the links that LSE is "holding" into a folder's white space in file managers other than Explorer. Not sure what you can do about this as Explorer replacement programs don't seem offer extensibility features on their "white space" pop ups. I've put a "warning" on this in the documentation. I actually can't imagine anyone who might use LSE would not be using a multi paned, multi tabbed file browser, but I could be wrong.

I suggest some cosmetic but important changes to the two step "Pick Link Source/ Drop HardLinks Here" function need to be considered

- Firstly the term "Pick Link Source" is confusing. I expected that selecting the option from the Action pop up would lead me into a dialogue where I would "pick" the file(s) that I wanted as the source of the link(s) I was about to create. I even wondered where the ellipsis (...) was, indicating that selecting Pick Link Source would lead into such a dialogue. I also wondered why the file(s) currently selected weren't the Link Source(s). When I selected "Pick Link Source" there was of course no resultant dialogue, in fact nothing obvious seemed happen. However when I did it again the "Pick Link Source" option had disappeared - very confusing!! Of course I had not read the documentation at this stage - but who reads documentation anyway.

Semantically the phrase *Pick Link Source* has an implicit future tense, i.e. it is something I am going to do. It is not something I've done, which is in fact the case, I selected ("Picked") one or more files, and then pressed the Action button on the mouse. I suggest the term be changed to *Set Source Link(s)*, this conveys the message that I am commanding the software to set the Link Source to the selected file(s). Alternatively you might prefer Store or Save Link Source(s), the word Set was the first alternative that came into my mind, I'd be just as happy with Save or Store, all three are infinitely superior to Pick.

On reflection if it were my decision I'd use Save Source Links.

- There must be something that informs the user that they have a Create action pending, i.e. they've Saved the Source Links, but haven't Dropped it/them or Cancelled the operation. I suggest that after Saving the Source Links that a View Source Links option be available, irrespective of context. This would display a submenu which would be a list of the Source Links currently in the buffer. What action ought to be taken if the user Selects or Actions an item in this list is moot, e.g. if the Action button is clicked on an item in the list would the Open, Edit, Play in Winamp ..., Properties etc. menu popup. My suggestion is that the list be non-selectable and non-actionable, i.e. it's display only, however I suspect the purists might disagree.

The View Source Links option would also be shown in the context where the Drop HardLink(s) and Cancel Pick Links actions shown. I suggest that Cancel Pick Links, be changed to Cancel Create Link(s), and that it be available irrespective of context, rather than only being available at a potential drop point. Thus one would always have the View Source Links and Cancel Create Link(s) choices available when there are Source Links in the buffer.

It occurs to me that I may be using this feature in a manner that the developers may not have envisaged. What I find attractive about the select/save/drop sequence is that I can do something useful between the select/save and the drop actions, like create or make the target folder visible, in doing so I might get distracted or diverted, having the View ... and the Cancel ... options in front of me acts as a reminder to finish that task.

Non NTFS File Systems

Does LSE work on file systems such as SAMBA or the upcoming Open Source NTFS Clone? I'm not suggesting it should, I was just wondering about whether I could create Junctions to an IBM RISC server running SAMBA under AIX. I won't explain why I might want to this.

HardLinks Across Volumes

If I Action drag a file from my F: drive to my Desktop (C:\Documents and Settings\Phil Daniels\Desktop), the HardLink Here choice is offered, but does nothing if selected. I would prefer the decision be made earlier and the HardLink Here choice not be shown if the drag crosses volume boundaries.

Offering users invalid choices that resolve themselves by doing nothing when they are selected is bad practice. It inevitably leads users into reporting false errors, e.g "HardLinking is broken because when I drag a file from my F drive to my Desktop and select HardLink Here nothing happens".

If I set up a Virtual Drive T: to "C:\Documents and Settings\Phil Daniels\My Documents\Taxation Documents\" then I can't get a HardLink on my Desktop (C:\Docs...\Desktop) to the file "T:\2005 Income Statement.xls". I can create a HardLink if the source selected is "C:\Documents and Settings\Phil Daniels\My Documents\Taxation Documents\2005 Income Statement.xls". This is exactly what I would expect to happen, after all Virtual Drives are intended to be transparent, I just thought you'd like to know.

Obviously I would not use a HardLink in this scenario, I'd use a a Shortcut, which the Link Tracking Service will maintain if I were to move my Taxation records to another location, including another drive.

Auto Rename

I think we have a philosophical difference here, when I create hard links I do not have any concept of there being "an original" file, however that concept is implicit in your Auto Rename feature. It is **not** implicit in NTFS's implementation of hardlinks, as you clearly show in your discussion of the deletion process (i.e. the count down on the reference count).

I currently have about 25,000 data objects, at a particular point in time (initial creation, which is an import process from various sources), an object has just one link. However the process of registering the new data object creates 3 additional links, one of which will be in a directory (not the same one every time) that is the target of at least two junctions, so each data object has a minimum of 6 effective links (4 real and 2 virtual), after about a month an average object will have 14 links (9 real, 5 virtual). I thought I'd give you these statistics so that you have some idea of the scope of what I'm doing with hardlinks and junction folders.

Back to your Auto Rename feature why limit it to one extra link in the originating folder, why not two or twenty two, and why only the originating folder, why cant I drag the same file repeatedly to the same destination and have "HardLink(1) to blah blah.txt", "Hard Link(2) to blah blah.txt" etc. And why can't I choose to have "HardLink to " prepended to all HardLinks I create, just as I can choose to prepend "Shortcut to " onto all my shortcut (.lnk) files.

I guess my message here is why implement something in such a limited fashion. I cannot imagine a scenario in which I would want two links to the same object in the same folder, but I admit to being very focused on the way I use HardLinks. NTFS Link allows the creation of multiple links to the same object, offering a similar Auto Rename service, with the addition of an index, as is implemented in LSE.

As a minimum the software should at least tell me I'm trying to drop a link whose name already exists in the folders directory, and give me the same options I would get if I were attempting to move or copy the file into the target folder.

My preference would be that I could disable duplicates at the folder level, what I'm suggesting that LSE implement some additional columns for folders

- | | |
|---------------------------|---|
| a) Duplicates: | 1 - Allowed,
2 - Disallowed,
3 - Ask User |
| b) Duplicate Name Format: | 1 - HardLink(n) to filename.ext,
2 - filename.n.exe,
3 - filename.exe.n |
| c) Duplicate Index Length | {1-6] |

I'm not sure how one goes about adding extra columns to folders, or even if one can do such a thing. If I was implementing it I'd attach an extra data stream to the folder with the settings specified in xml, the only development tool I have is Jscript, so I'm limited by its capabilities.

This issue of course is predicated on file names being unique within a folder's directory. Each link to a data object can have a different name in each of the folders in which the links reside. There is nothing to prevent me dropping each link into another folder, which would then have a multiple directory entries with different names each linking to the same object.

Junction Folders

In the LSE Document an attempt was made to draw a parallel between Junction Folders and HardLinks. This is very dubious, as you would realise that HardLinks and Junction Folders are very different animals indeed.

A significant issue is that I can't find a way to use LSE to create a Junction Folder with a name that is different to that of the Source Folder. As you probably know one cannot use normal renaming methods on Junction Folders any more than one can use normal deletion methods, the behaviour of Explorer is in fact the same – it crashes.

One of the more common usages of Junction Folders is to coerce Windows and other programs into allowing software to reside somewhere other than "C:\Program Files\". NTFS allows an extant vanilla folder to be changed into a Junction Folder, i.e. the folder must be empty. Thus if one has an empty folder "C:\Program Files" (which is no mean feat) then one can Junction it elsewhere, in my case it's junctioned to "f:\software\bin\". Using LSE when I drag "f:\software\bin" on to C:\Program Files\, a junction folder with the path C:\Program Files\software\bin is created, what I want is a junction folder called C:\Program Files\.

I guess I could rename f:\software\ bin to f:\software\Program Files\ and drop it on Drive C:, I've never tried to create a junction by dropping the target onto a drive. I just did that, it wasn't Program Files, I already have that set up, the junction was created. I've never tried renaming the target of a junction, so I'll try that. I got a warning from NTFS Link telling me that the folder was the target of a junction folder, it gave me the option of fixing up the junction(s) reference column or cancelling, I chose the former. Strange I tried it again and I did not get the warning from NTFS Link, hence the link was broken. Thanks to LSE I was able to easily delete the junction. Now I can't create any junctions, I've had this problem before, the only way to resolve it is to reboot.

I'll do some more testing after uninstalling NL and LSE and reinstalling LSE.

I also use junction points to invert folder hierarchies.

In simple terms assume I have a year->name folder structure that I want to invert into a name->year structure, so the data in folder "\2005\fred" is also available from the folder "\fred\2005\". Sounds simple enough, and it is if I use software that allows me to transform empty folders into Junction folders, alas I cannot seem to be able to do that with LSE.////

Another problem I'm having with Junction folders is that whilst HLSE and NL happily coexist with respect to HardLinks, they "clash" when it comes to Junction Folders. I haven't looked into the cause in detail, but I'm pretty sure they are competing for one or more registry keys. It seems to be the case that whichever of the two products was installed last, more or less works; whilst only limited features of the product that was installed first continue to function..

At the moment, because I reinstalled NL after installing LSE, the only LSE Junction facility that seems to be fully functional is the Junction Delete function. I want to let you know that it can delete Junctions created with NL, Junction Magic and my adaptation of Mark R's junction program. Well done, yours is the ONLY shell program I know of that can reliably delete junction folders no matter how they were created, up until now I religiously used Mark R's junction command line program to delete junction folders, now I can do it with my mouse!

To overcome my inversion problem I suggest the following, if a single folder is dropped into or onto an EMPTY folder, then the user is given the choice of making that folder the junction folder. Thus the Drop Here ... sub menu would have three options, "Junction In Here" (i.e. what currently happens), "HardLink Clone In Here" and "Junction This Folder".

Finally the "small piece of chain" is only apparent in the Large Icon view, which I only use on rare occasions, my default view mode is Detail. In the situation where you create the junction folder you might like to consider using the icon I use for all my junction folders, as you appreciate it's really important that one knows one is looking at a Junction Folder as opposed to a Ordinary Folder.

BTW In your HTML document I think you wrote that if an attempt is made to delete a Junction folder via normal means in Windows Explorer all the files in the target folder are deleted. In Windows XP this has never happened to me, and I've made the mistake of using the delete key on them many times. What happens is that Explorer or xplorer² crashes, Dr Watson crashes, XP shakes its head and feathers, reloads Explorer, icons disappear from the System Tray, even though the applications are still running and one carries on, the Junction is still there, as are all the files in its target.

HardLink Whereabouts

One of the problems with using HardLinks is knowing where they are, the Reference count informs one of their number but not of their whereabouts.

I have overcome this problem by implementing a ":LinkPaths" stream on files with multiple hardlinks. The paths to the various links are maintained in this stream. However this requires me to ensure that I only use my file manipulation functions (linkcreate, linkdelete, linkmove, linkcopy and linkrename) on such files. I don't know how to replace the standard XP services, and even if I did I probably wouldn't.

My functions are implemented in JScript, and invoke command line facilities to create the hardlinks (fsutil) and StreamTools CS program to unload the :LinkPaths stream into a temporary file so I can manipulate it in Jscript code and then again to copy the temporary back into the ":LinkPaths" stream. All this is terribly inefficient, but it does work.

What I'm looking for is a software developer who is interested in implementing this functionality in compiled code and if possible in a manner that preempts, pre-processes, post-processes standard Windows functionality, so that standard file manipulation functions (New, Copy, Move, Delete etc.) can be used.

Let me know if you're interested in pursuing the idea further. If you are then I will give you 3 months exclusive access to the Jscript code and my intellectual assistance in implementing the scheme (don't expect me to write code), including testing. If you have a usable product at the end of that time then you own the IPR, all I'd ask for is a byline in the credits, and free access to the resultant software product. If you fail to produce something useful within three months then I have the right to pass on the idea and my code to other s/w vendors.

I'm not interested in making money out of this idea, I just want to see it implemented.