Link Shell Extension

Last Updated August 23th 2013, Version 3.7.5.1

Introduction

The NTFS file system implemented in NT4, Windows 2000, Windows XP, Windows XP64, and Windows7/8 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 be created with the POSIX command ***ln*** included *in* the Windows Resource Kit, the *fsutil* command utility included in Windows XP or my command line ln.exe utility 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 , Volume Mountpoints , and Windows7/8's Symbolic Links, (herein referred to collectively as Links) a folder cloning process that utilises Hardlinks or Symbolic Links and a copy process taking care of Junctions, Symbolic Links, and Hardlinks. 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 XP64 and Windows7/8. Hardlinks, Junctions and Symbolic Links are NOT supported on FAT file systems, and nor is the Cloning and Smart Copy 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.

LSE is installed by executing the install program (HardLinkShellExt_$(platform).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\LinkShellExtension

During installation Explorer.exe has to be restarted to make Link Shell Extension active. This means that all pending operations with explorer.exe are interrupted, but with the interactive install you can decide to postpone the explorer.exe restart. A dialog box will give you the choices during installation.

Silent Installation

LSE can also be installed silently by specify the /S switch and an argument for the language from a command prompt. e.g.

```
HardLinkShellExt_$(platform).exe /S /Language=English
```

Currently *English, Chinese, Czech, French, German, Italian, Polish, Portuguese, Russian, Slovak, Spanish, Swedish and Turkish* are available as valid parameters for the /Language switch.

If the /S switch is used, explorer.exe will be restarted after installation, to make Link Shell Extension active immediately.

When using the silent install a directory can also be specified with the /D switch e.g.

```
HardLinkShellExt_$(platform).exe /S /Language=English /D=C:\Program Files\LSE
```

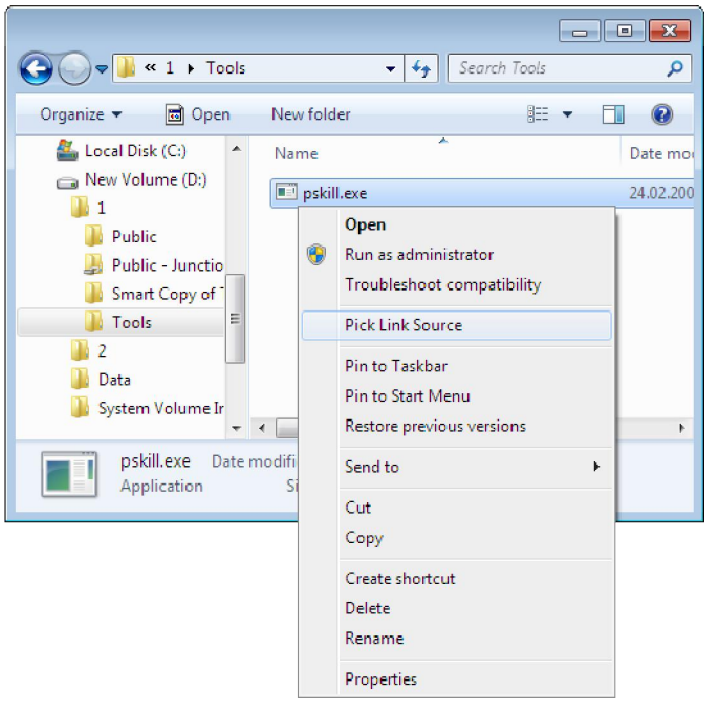
Link Shell Extension can also be uninstalled silently by issuing

```
$LSEInstallDir/uninst-HardLinkShellExt_$(platform).exe /S
```

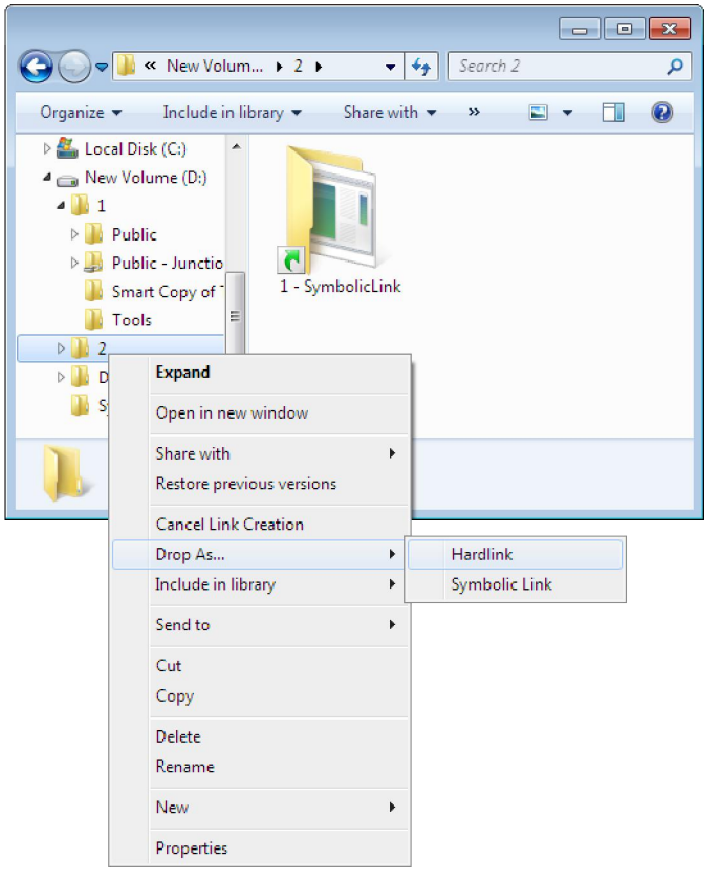
If the /S switch is used, explorer.exe will be restarted after uninstallation, to make Link Shell Extension inactive immediately.

Using Link Shell Extension

Pick Link Source causes 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**

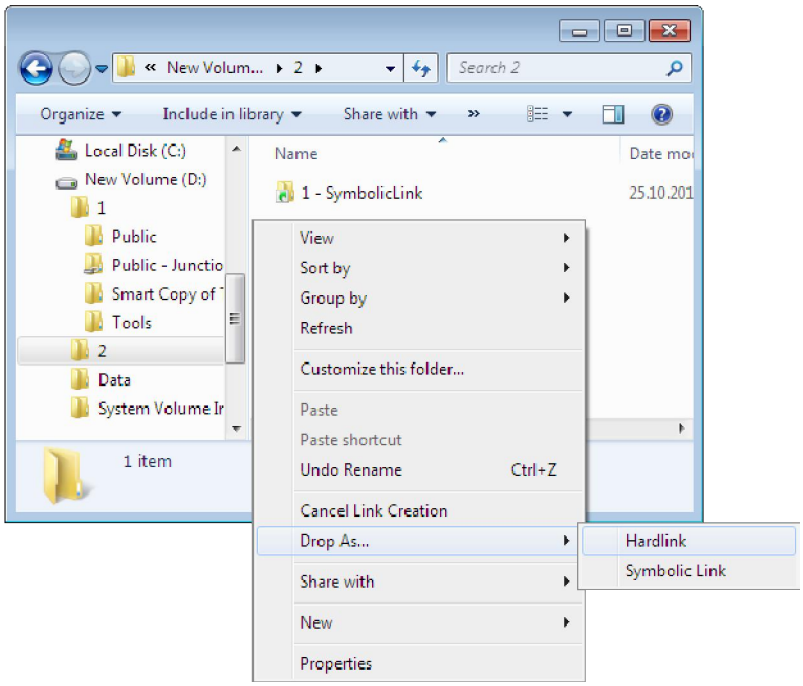


Choosing **Drop HardLink** will create the Hardlinks in the selected destination folder.

Overlay Icons for Hardlinks
To help distinguish hardlinks folders from normal files, an **overlay icon** is implemented on hardlinks that shows a red arrow icon under the folder.

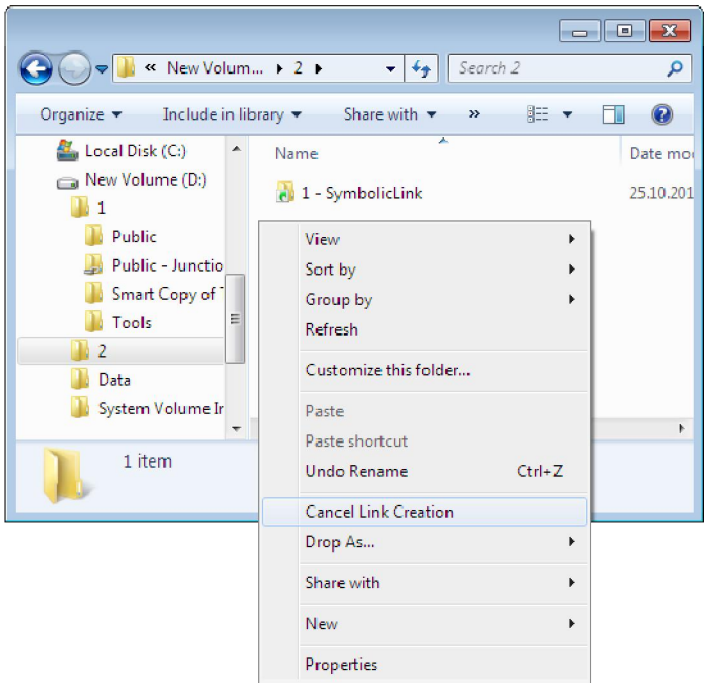
Overlay icons for Hardlinks can also be customized.

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.



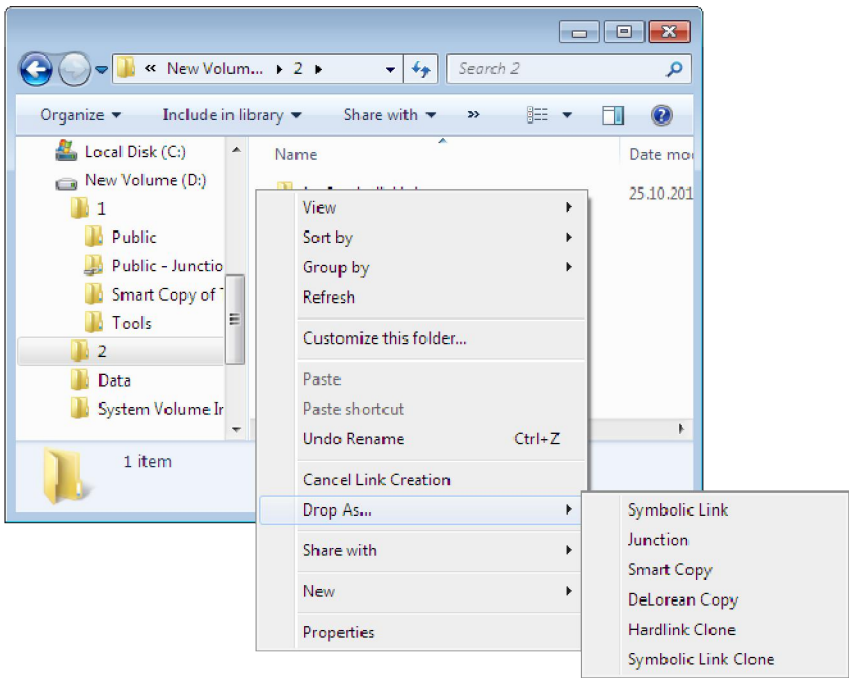
Cancel current Pick Link operation

When doing an Action button click in the destination folder background, in addition to the Drop HardLink option there is the possibility to **Cancel Link Creation** entry.



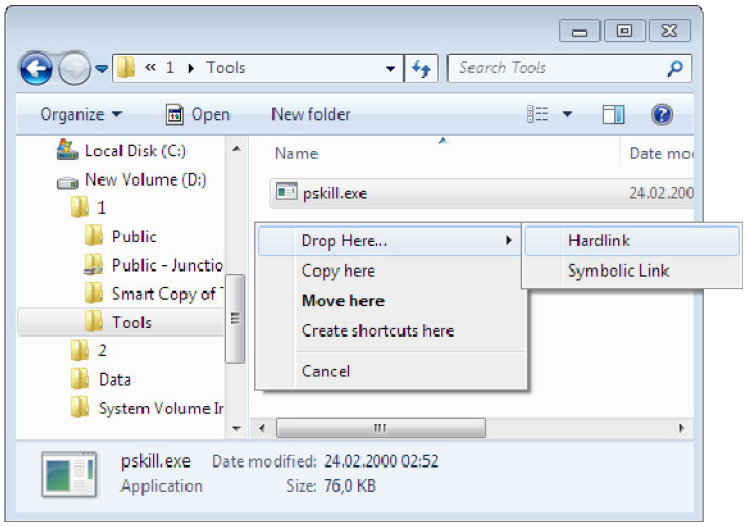
Popup Submenue Since LSE supports Junction, Clones and with Windows7/8 Symbolic Links, when one or more folders are selected as the Source Links they can be dropped in several forms.

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



Drag and Drop Support

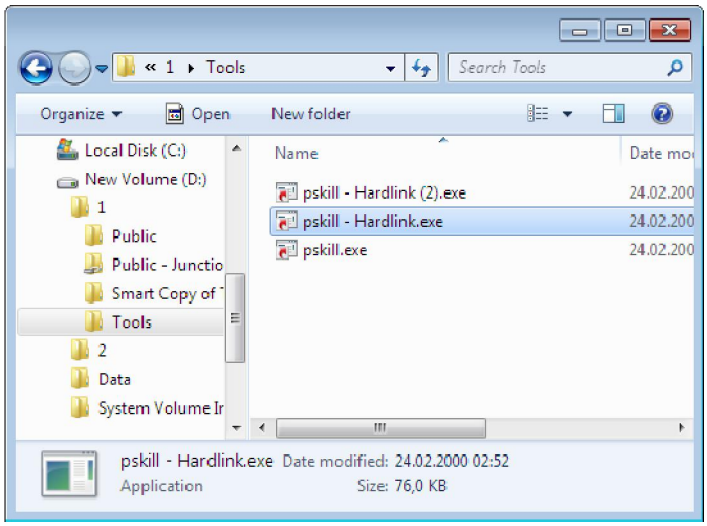
Creating Hardlinks via drag and drop is supported, after selecting one or more files you can drag them to the destination folder with the Action button held down; when it is released choose **HardLink Here** from the action menu to create the Hardlinks of the selected files in the destination folder.



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 '$filename - Hardlink.$ext' as the name of the the new link.

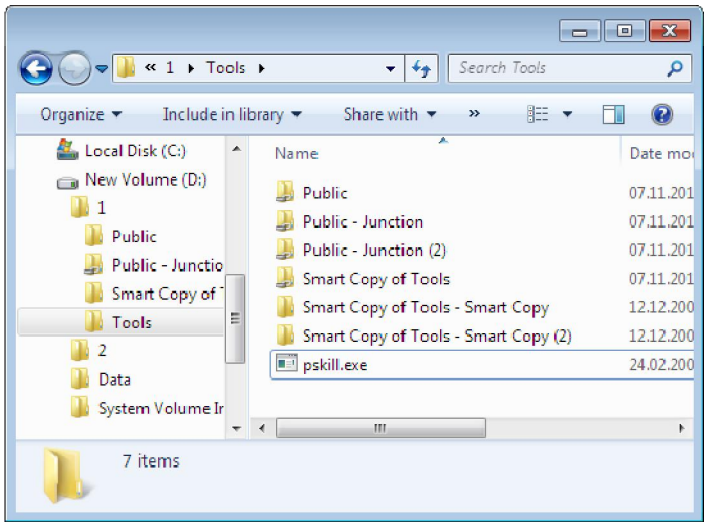
With Windows XP this behaviour is different and LSE behaves under XP accordingly: 'HardLink of $filename'



LSE uses the same hydraulics as explorer when it comes to multiple '$filename - Hardlink': It uses numbers to enumerate the multiple hardlinks of one file in the same directory, e.g. *$filename - Hardlink (2).$ext.*

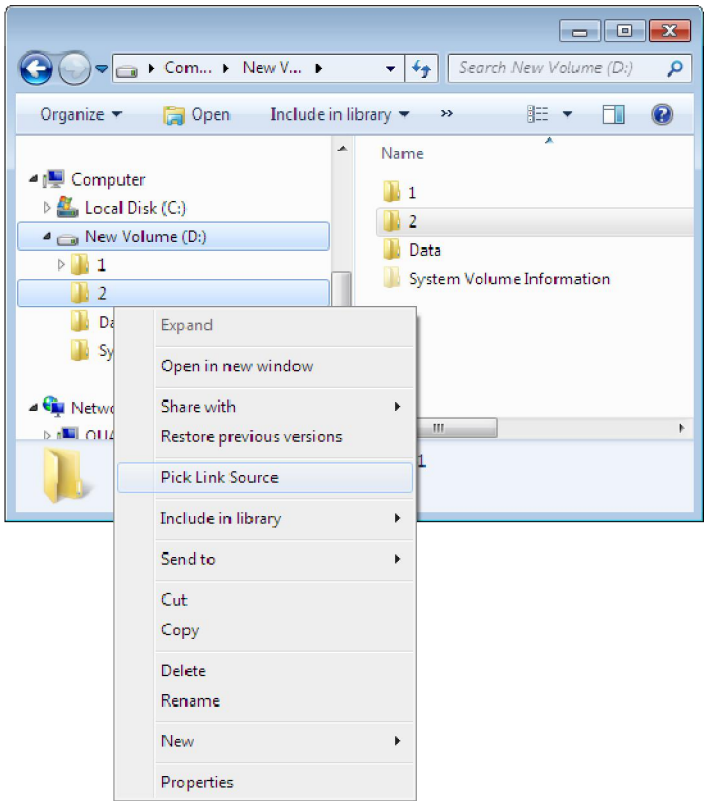
With Windows XP this behaviour is different and the names are generated like *'Hardlink of $filename (2).*

The Auto Rename mechanism is also used when Junctions, Hardlink Clones, Symbolic Links, Symbolic Link Clones, Mountpoints or Smart Copies are created in the same directory.

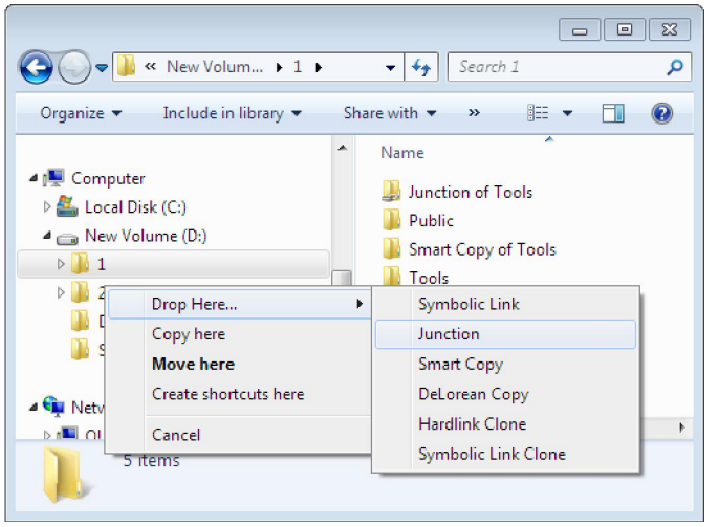


Junction Support

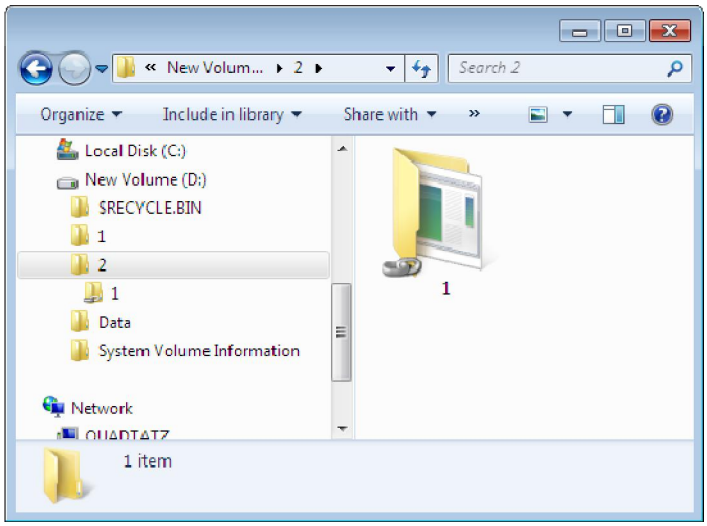
Junctions are a feature of NTFS version 5.0, they provide for the creation of linkages among directories, Junction were not supported in NTFS Version 4.0



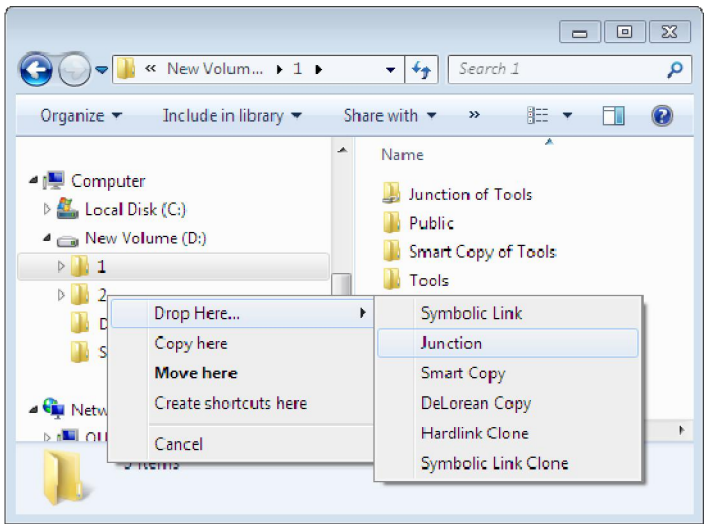
Junctions are created in the same way as Hardlinks, except that the Source Link is a folder rather than a file. Select a folder, click the right mouse button, choose **Pick Link Source** from the action menu, navigate to the destination folder, click the action button, open the submenu **Drop As ...**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



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



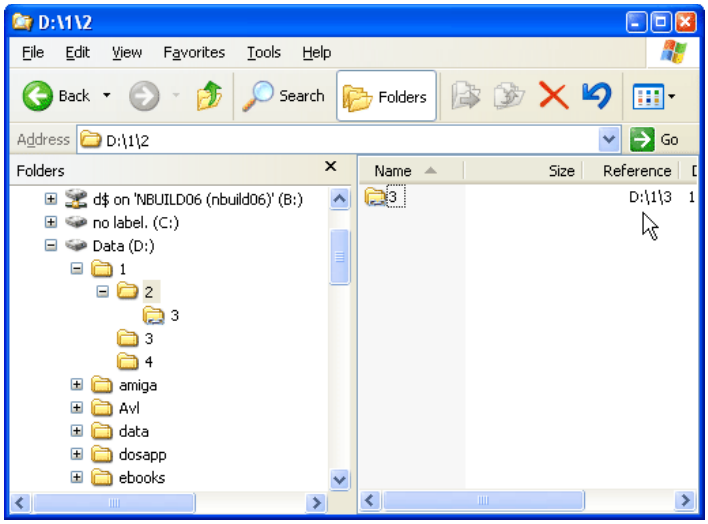
Junctions can also be created via Drag and Drop when the selected folders are dragged with the action button pressed to a destination folder; when the right mouse button is released, select the **Drop Here ...** submenu and then **Junction.**



Junctions can be deleted by using the Delete commands from Explorer as usual, if Link Shell Extension is installed, because Link Shell Extension implements a so called CopyHook handler, which intercepts Explorers Delete commands, and thus fixes Explorers problems with junctions.
Link Shell Extension changes Delete commands from Explorer on Junctions to unlinking Junctions, and not deleting the content within a junction, which seems most logical.

Windows7/8: With Windows Vista & Windows7/8 explorer is natively aware of Junctions, and no interception or CopyHook handler is necessary, Explorer simply works out of the box with junctions. This is also the reason why the *Delete Junction* menue does not show up on Vista and Windows7/8.

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



Junctions are a powerful feature, but can also be a **dangerous feature** without Link ShellExtension precautions under Windows Version < Windows Vista. Nativly Explorer shipped with < Vista 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.

Overlay Icons for Junctions
To help distinguish junction folders from normal folders, an **overlay icon** is implemented on junctions that shows a small three link chain icon under the folder.

Overlay icons for junctions can also be customized.

Junctions can **span network drives** as long as the target is a mapped network drive. Unfortunately Junctions, which have a UNC Path as target, can be created with LSE, but even Vista seems to contain a bug, which prevents it from dereferencing a UNC Path in a junction, even if LSE correctly sets up the reparse info for UNC junctions. When a UNC target junction is double clicked in explorer the error ERROR_INVALID_REPARSE_DATA(4392), will show up and tell you that the info in the reparse point is illegal, even if it is not.

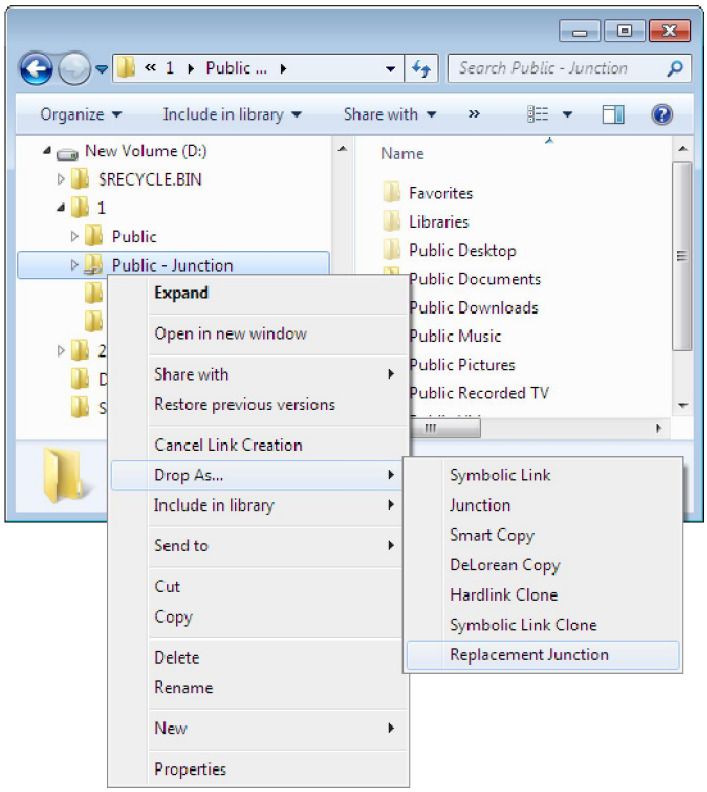
@Microsoft: Why didn't you enable this feature for junctions, even if the syntax for UNC junctions is defined: \??\UNC\server\share. Any help appreciated.

With Vista and Windows7/8 some folders, e.g. c:\Program Files needs elevation for junctions beeing created in. This is why the famous UAC dialog must be acknowledged.
To be exact: Only the creation of directories needs elevation in such situations, but creating an empty directory is a vital part of creating a junction. The DeviceIoControl(), which does the real work in creating junctions would work without elevation.

Replacement Junction Symbolic Link Mountpoint

Link Shell Extension can change the target of an existing Junction, Symbolic Link or MountPoint either via Pick/Drop or Drag and Drop.

To use this feature simply select an existing directory as Link Source and drop it over an already existing Junction/Symbolic Link/Mountpoint. By selecting the 'Drop as ... Replacment Junction/Symbolic Link/MountPoint' from the action menu, the target of an already existing Junction/Symbolic Link/MountPoint is replaced by the new picked target.

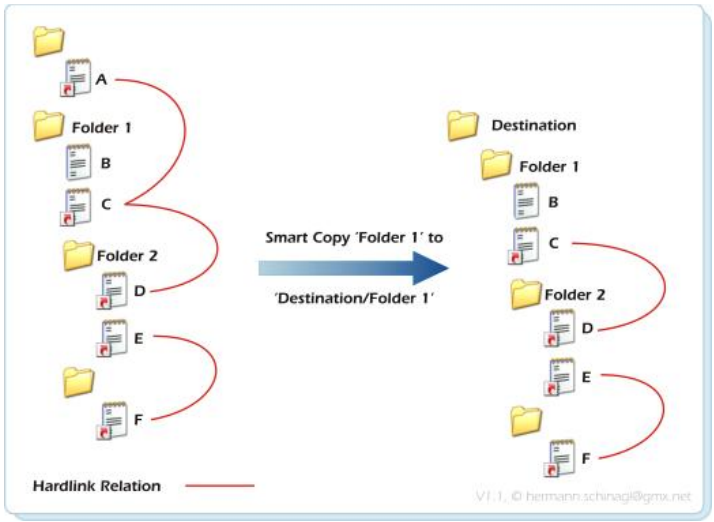


The same can be achieved via Drag an Drop.

Replacement functions can also be used to repair broken junctions/symbolic links/mountpoints.

Smart Copy Smart Copy basically creates a copy of the directory structure from the source location to the destination, but it preserves the inner hardlink structure and inner junction/symbolic link relations of the source, and recreates this inner hardlink structure and inner junction/symbolic link relation at the destination location:

With hardlinks it behaves as follows:



By closely looking at the above picture one can find three different types of files:

Normal Files The file B is a normal file. It gets copied as any other copy tool would do.

Saturated Hardlinks The files E and F are hardlinked together. In LSE naming universe they are called *Saturated Hardlinks*, because the reference count, which is here 2, matches the number of occurrences below 'Folder 1', which is here 2.

In General: A hardlink is called *Saturated* with respect to a folder *F*, if the number of occurrences below the folder *F* matches the reference count.

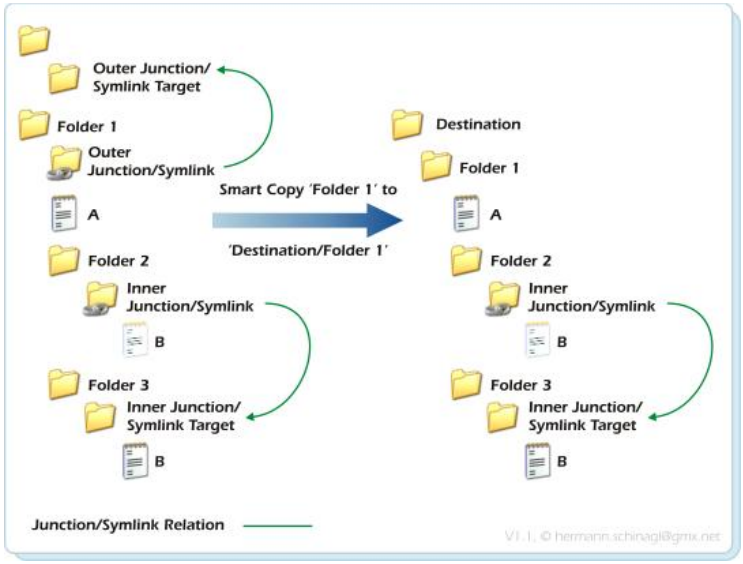
Unsaturated Hardlinks

Saturated Hardlinks can be copied completely via Smart Copy.
The File A, C, D are hardlinked together. In LSE naming universe they are called *Unsaturated Hardlinks*, because the reference count, which is here 3, does not match the number of occurrences below 'Folder 1', which is here 2. Only C and D are below Folder 1.

In General: A hardlink is called *Unsaturated* with respect to a folder *F*, if the number of occurrences below the folder *F* is smaller than the reference count.

Unsaturated Hardlinks can only be partially copied by Smart Copy. In the above example C and D are hardlinked together in the destination location, but the hardlink to A is broken. This means that the reference count of C and D is 2 with the destination location.

With junctions or symbolic link directories the default behaviour during smartcopy is as follows:



By closely looking at the above picture one can find three different types of folders/junctions:

Normal Folders The folder 'Folder 3' is a normal folder. It gets copied with its content as any other copy tool would do.

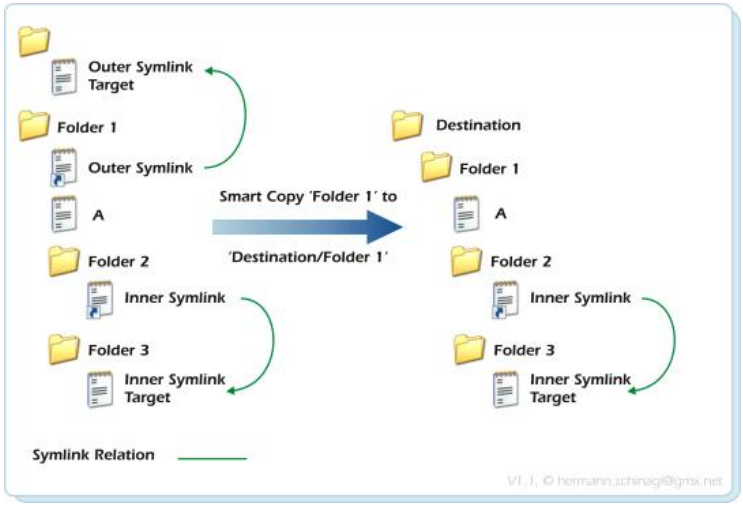
Inner Junctions Symlinks The folder 'Inner Junction/Symlink' is targeted at 'Inner Junction/Symlink Target'. In LSE naming universe this kind of folder is called *Inner Junction/Symlink*, because its target points to a folder, which is below the common anchor 'Folder 1'.

*Inner Junctions/Symlink*are restored properly via Smart Copy in the destination location.

Outer Junctions Symlinks The folder 'Outer Junction/Symlink' is targeted at the folder 'Outer Junction/Symlink Target'. In LSE naming universe this kind of folder is called *Outer Junction/Symlink*, because its target points to a folder, which is in paralell and thus outside the anchor 'Folder 1'.

Outer Junctions/Symlink can be handled in three different ways. Please see the section on Outer Junction/Symlink Handling.

Windows7/8 and Windows Vista support Symbolic Links, which behave as follows during Smart Copy:



By closely looking at the above picture one can find three different types of files/symbolic links:

Normal Files The file A is a normal file. It gets copied as any other copy tool would do.

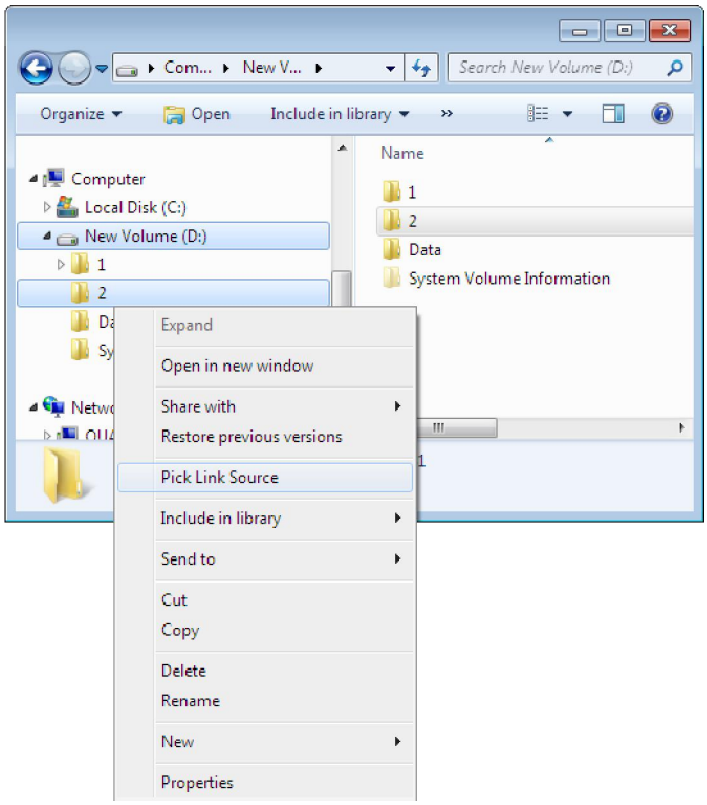
Inner Symbolic Links The symbolic link 'Inner Symlink' is targeted at 'Inner Symlink Target'. In LSE naming universe this kind of symbolic link is called *Inner Symlink*, because its target points to a file, which is below the common anchor 'Folder 1'.

Inner Symlink are restored properly via Smart Copy at the destination location.

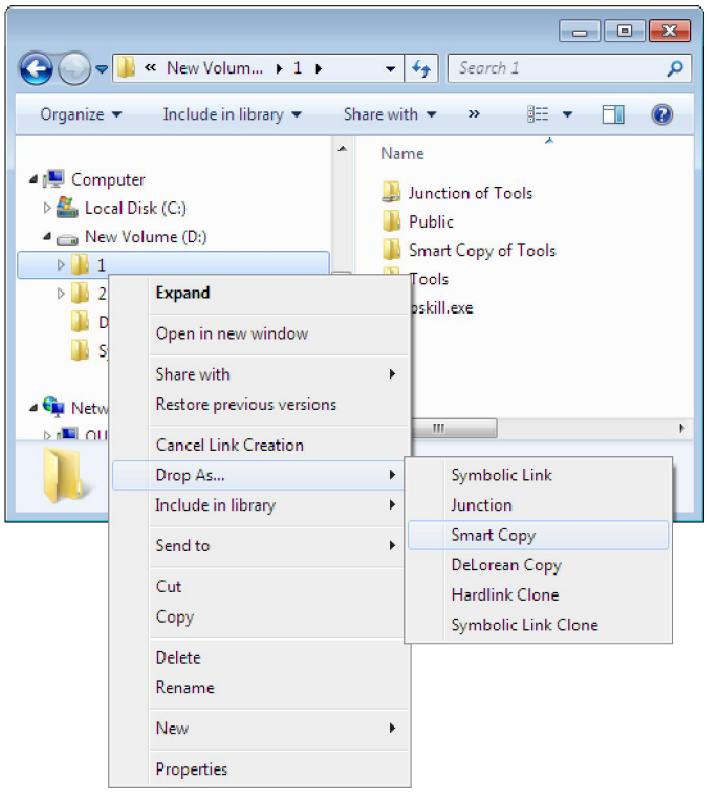
Outer Symbolic Links The symlink 'Outer Symlink' is targeted at the folder 'Outer Symlink Target'. In LSE naming universe this kind of symbolic link is called *Outer Symlink*, because its target points to a file, which is in paralell and thus outside the anchor 'Folder 1'.

Outer Symlink are handled by Smart Copy depending on the Outer Junction/Symbolic Link handling.

Smart Copies are created in the same way as Junctions, select a folder, click the Action button, choose **Pick Link Source** from the action menu...



...navigate to the destination folder, press the action button, open the **Drop As ...** submenu and select **Smart Copy**:



Smart Copy is a must if e.g.. the whole content of a hard disk, which has lots of hardlinks/junctions/symbolic links, should be copied to another hard disk. During the Smart Copy operation empty folders get copied too and the date/time stamps of folders/junctions/symbolic links are also restored at the corresponding destination locations.

Because Smart Copy creates inner hardlinks/junctions/symbolic links, this feature is only available on NTFS volumes.

If Smart Copy takes longer than 250msec a progressbar shows the status of the smart copy operation.

Smart Copy also processes all available alternative NTFS streams of a file.

Currently ACLs are not copied with Smart Copy, but it is on LSE's todo list.

When restoring Symbolic links under Windows7/8 and Windows Vista, LSE forks its helper symlink.exe to forwards this operation to it, because the creation of symbolic links needs elevation, and thus brings up the famous UAC dialog.

LSE **only** issues its helper symlink.exe if a symlink is among/below the selected folders, so it saves you from one UAC prompt if you don't have symlinks among your selection.

Smart Copy creates relative symbolic links during the Smart Copy operation.

Command Line

The Smart Copy functionality is also available via command line from ln.exe via the --copy command line switch.

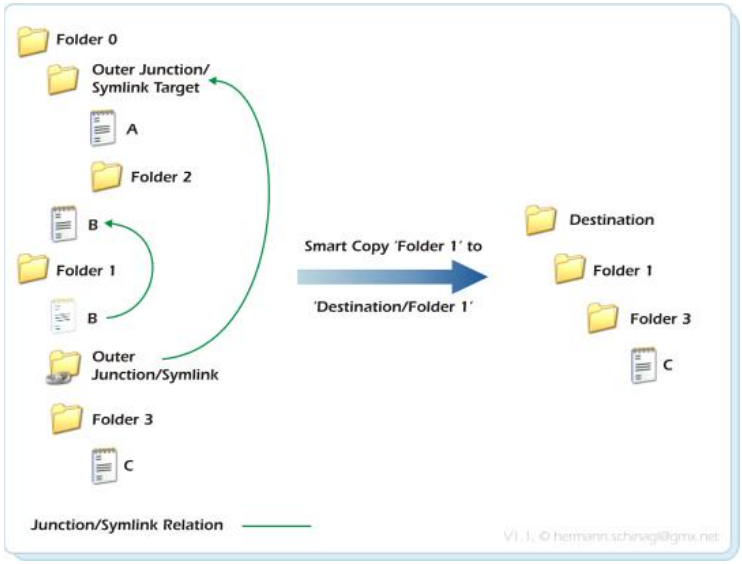
Crop/Unroll/Splice Outer Junctions/ Symbolic Links

During SmartCopy, Smart Mirror, Delorean Copy and Clone so called Outer Junctions/Symlink directories may need processing. There are 3 different ways to deal with those Outer Junctions/Symlink directories:

Crop

Crop breaks links to Outer Junctions/Symlink directories in the destination.

Crop also applies to Outer Symlink Files.



In the above example *Folder1* is copied to *Destination/Folder1*, but *Outer Junction/Symlink* is not available in the destination, because *Folder1/Outer Junction/Symlink* pointed to *Folder0/Outer Junction/Symlink Target*, which is not below *Folder1*.

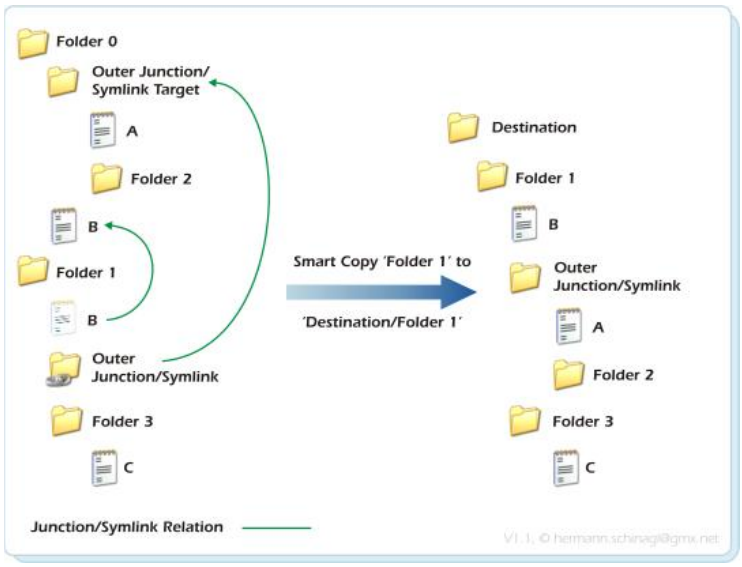
The objective behind cropping Outer Junctions/Symlink Directories is to get a pure copy during Smart Copy, Smart Mirror, Delorean Copy and Clone without connections to the source.

Enabling Crop for Outer Junction/Symbolic Links
Crop can be selected via the configuration tool.

Unroll

Unroll follows Outer Junctions/Symlink Directories and rebuilds the content of Outer Junctions/Symlink Directories inside the hierarchy at the destination location.

Unroll also applies to Outer Symlink Files, which means, that unroll causes the target of Outer Symlink Files to be copied to the destination location.



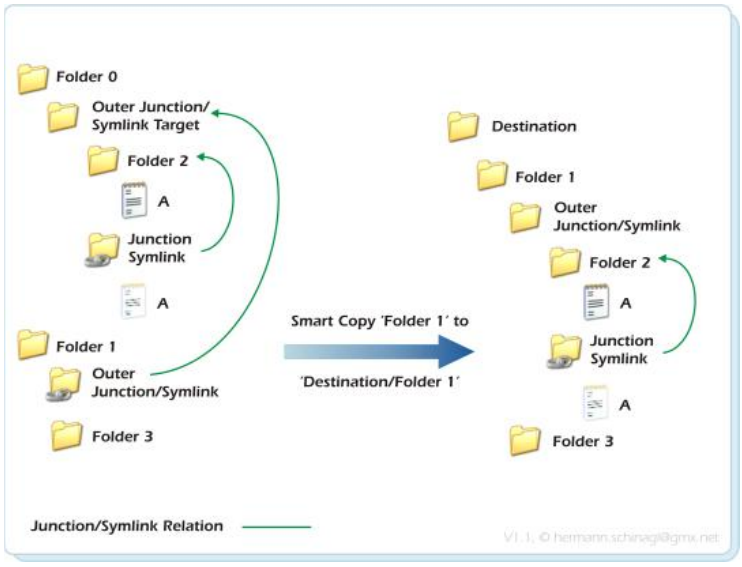
In the above example *Folder1* is copied to *Destination/Folder1*, and *Outer Junction/Symlink* and all the files/directories below *Outer Junction/Symlink Target* are copied to the folder *Outer Junction/Symlink* in the destination.

The objective behind unrolling Outer Junctions/Symlink Directories is to get everything with which the source is connected and rebuild it as seperate copy in the destination. It resembles the 'hair of the elephant' pattern: Pull on a hair of an elephant, and get the whole elephant.

Unroll is the default behaviour for Smart Copy, Smart Mirror, Delorean Copy and Clone.

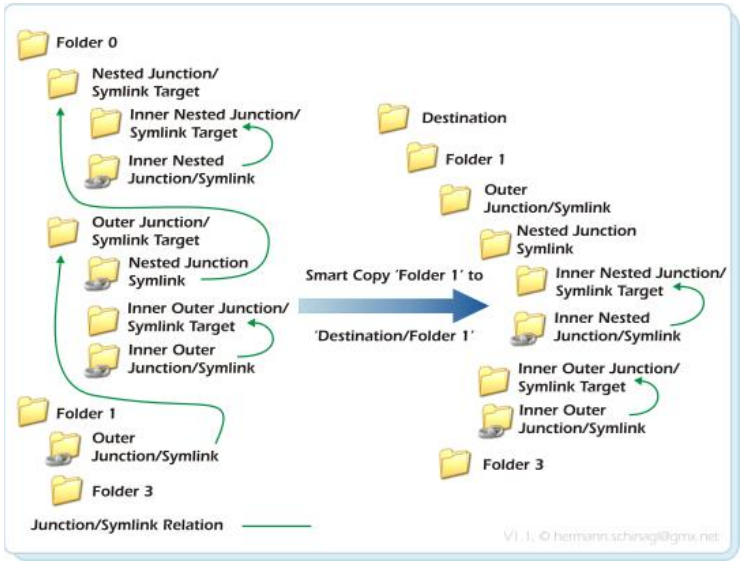
Advanced thoughts on Unrolling

The picture above was just the simplest case, because *Unroll* does much more when it encounters complex situations. Think of a outer junctions/symbolic links, which itself contains junctions/symbolic links, which are inner with respect to the first outer junction symbolic link:



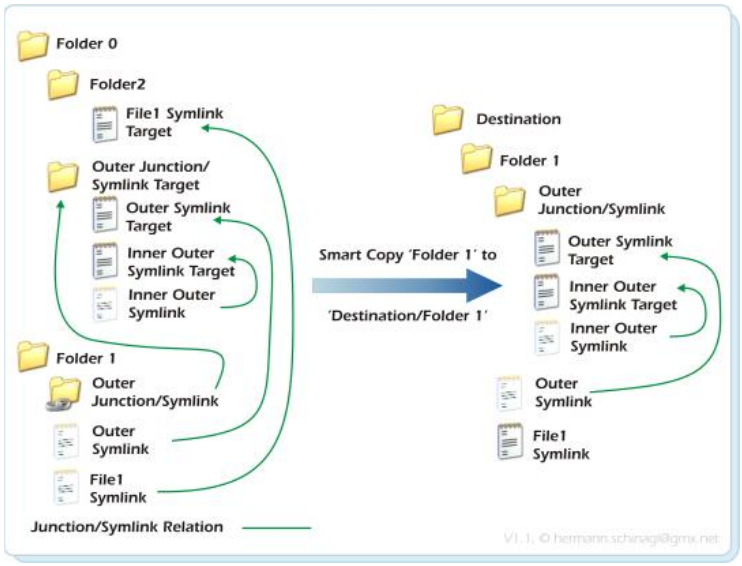
In the above example *Folder1* is copied to *Destination/Folder1*, and *Outer Junction/Symlink* and is unrolled as expected, but since *Junction/Symlink* is a inner junction with respect to *Outer Junction/Symlink Target*, the junction/symlink relation is restored in the destination.

This kind of **nesting** can be much more complex:



In the above example *Folder1* is copied to *Destination/Folder1*, and *Outer Junction/Symlink* and is unrolled as expected, but then it starts to get fascinating, because we have two levels of outer junctions/symlinks which all have respective inner junctions/symlinks, and which are restored properly. Once you digged yourself through the above picture, you got it. It is not simple I know, but it is neccessary to properly unroll.

And complexity increases if **symbolic link files** are within unrolled outer junctions/symbolic links:



In the above example *Folder1* is copied to *Destination/Folder1*, and *Outer Junction/Symlink* and is unrolled as expected, but it contains *Inner Outer Symlink* which points to *Inner Outer Symlink Target* and this is a inner junction/symbolic link with respect to *Outer Junction/Symlink Target*

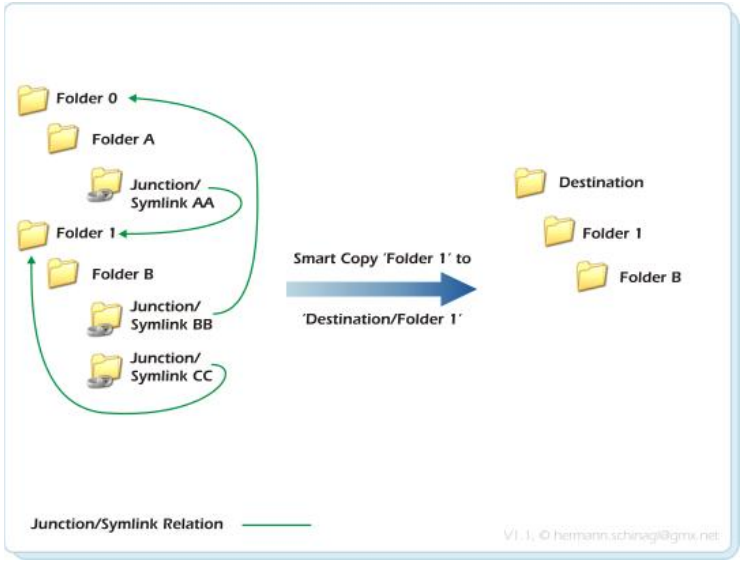
But worth mentioning is the Symbolic Link *Outer Symlink*, which would be a definitive outer symbolic link, but since its targets parent-directory *Outer Junction/Symlink Target* is unrolled, *Outer Symlink* becomes an inner symbolic link with respect to *Folder1*.

File1 Symlink is also a outer symbolic link, but its target parent-directory *Folder2* is not that lucky to get unrolled, so in the destination *File1 Symlink* is not a symbolic link any more, but a copy of the symbolic links' target.

Nested Reparse Points are also an interesting use case, which the algorithm has to tackle with:

In the above example *Folder1* is copied to *Destination*, and *Junction/Symlink F0* and is unrolled as expected, but it contains inner nested reparse points. *Nested* means Reparse point pointing to Reparse Points

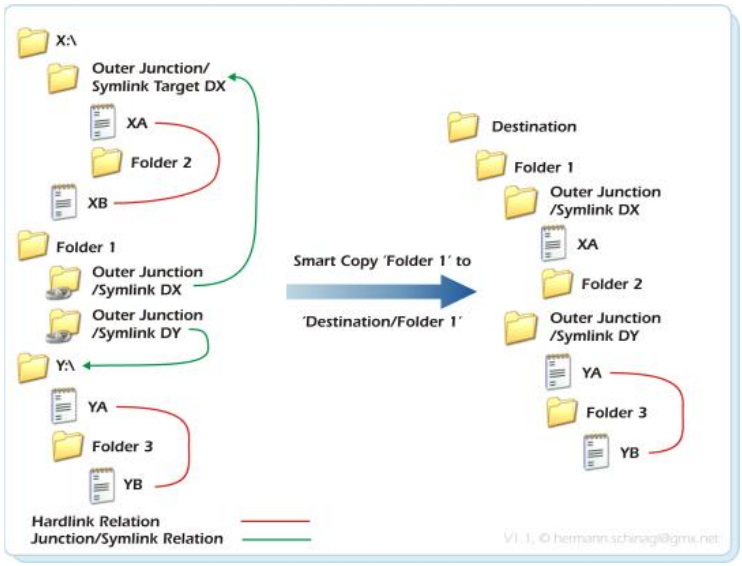
The Unroll functionality also opens up the possibility to have **circular** Junction/Symbolic Link relations among a set of copied items:



In the above example *Folder1* is copied via the --unroll option to *Destination/Folder1*. Smart Copy/Smart Mirror and Delorean Copy operations can deal with the above shown circularities and break circularities by not following the affected

Junction/Symbolic Link.

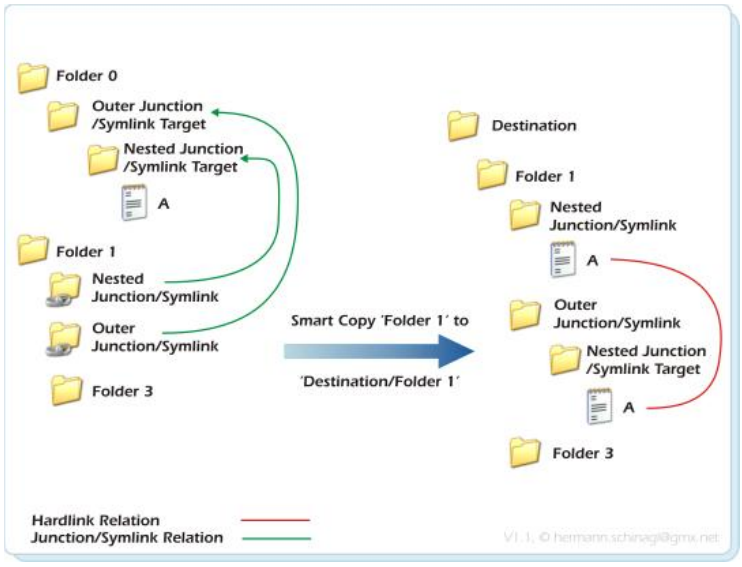
Junctions/Symbolic Links can also point to FAT drives or other NTFS drives requiring as a prerequisite unique Disk-IDs on all disks, which are chained together via Junctions/Symbolic links:



Hardlink siblings are found by matching the per NTFS volume uniqe file-id, but if more volumes are chained together it might happen that the same file-ids can be found on two different NTFS volumes. To address this all operations use the disk-id and the file-id to match hardlink siblings.

Furthermore it is not allowed to have the disk-id 0xffff-ffff, because the algorithms use this as internal indicator of a FAT drive.

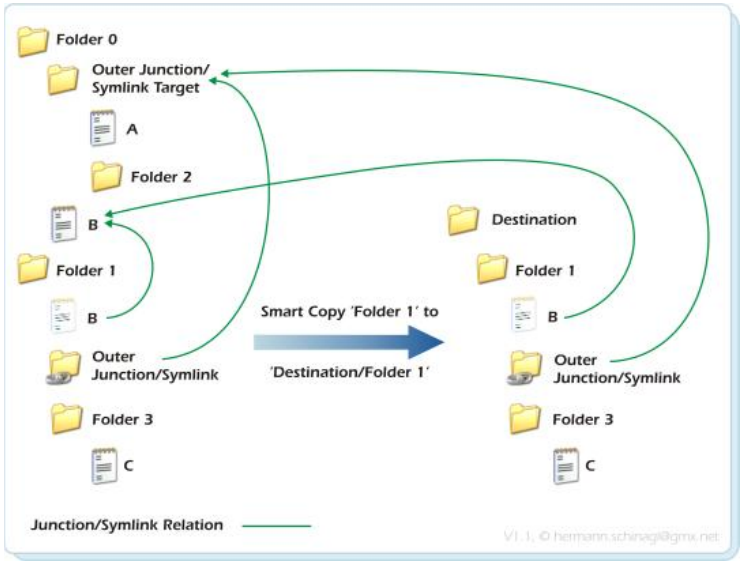
The Unroll option also allows to point multpile junctions to the same target location, which causes the algorithms to traverse the same items many times:



At the first glance multiple traversation of items looks simple, but for files this means that multiple traversed files are the same in the destination and are hardlinked together. So don't be confused when you see hardlinks, which have never ever been there before.

Splice

Splice reconnects Outer Junctions/Symlink directories in the destination to their original targets.



In the above example *Folder1* is copied to *Destination/Folder1*, and *Outer Junction/Symlink* is available in the destination as junction, which points to the original location *Outer Junction/Symlink Target*.

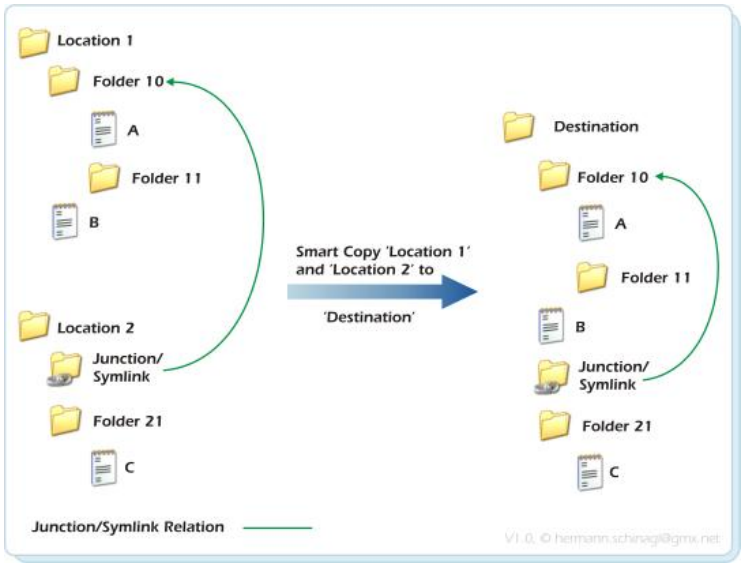
The objective behind splicing Outer Junctions/Symlink Directories to its original location is to get a copy during smartcopy, but to reuse Outer Junctions/Symlink Directories source locations.

Enabling Splice for Outer Junction/Symbolic Links

Splice can be selected via the configuration tool.

Multiple Source

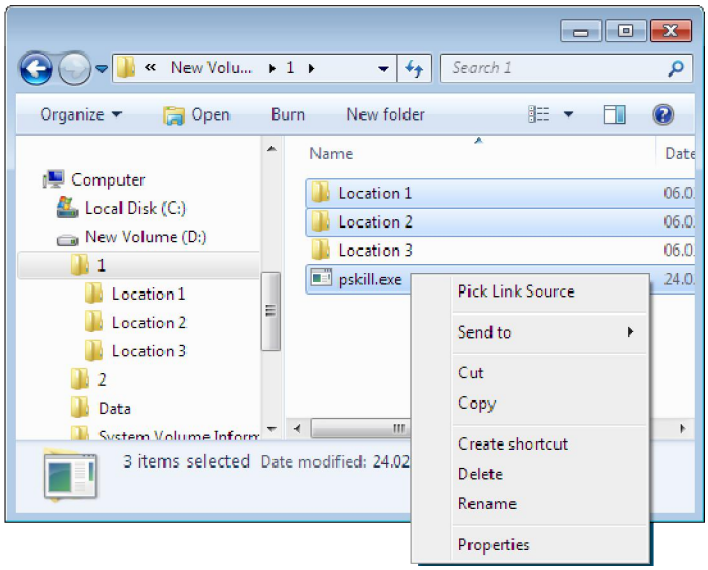
Multpiple Source Locations can be specified for SmartCopy, Clone, and Delorean Copy. If there are junctions/symlinks between these source locations, they are handled as inner junctions/symbolic links, because all source locations are dealt like a common root.



In the above example *Location1* and *Location2* are copied to *Destination*. *Location2/Junction20* is treated as inner junctions to *Location1/Folder10* in the source, and that's why *Destination/Junction20* points to *Destination/Folder10 in the Destination*.

The objective behind this is to treat all junctions/symlinks as inner junctions/symlins as long as they are in the set of source folders.

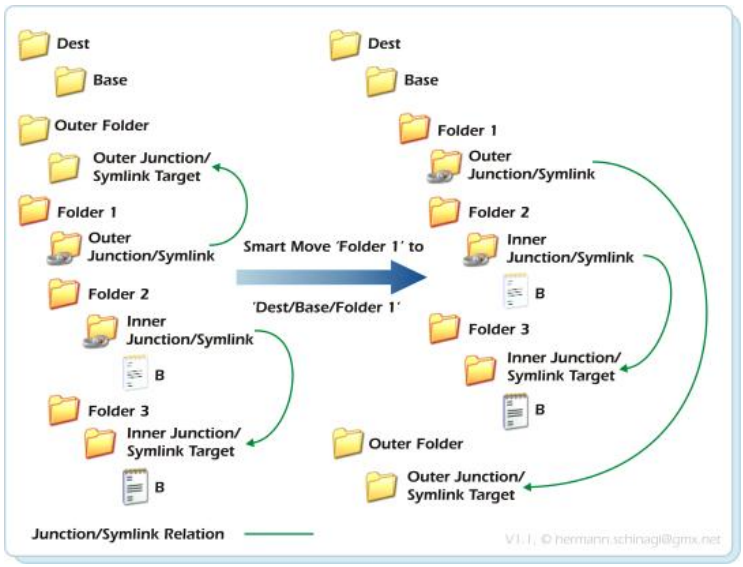
With LinkShell Extension this works as follows:



In the above example all content from *Location 1* and *Location 2* and pskill.exe are selected. Possible junctions/symbolic links in *Location 1* pointing to *Location 2* or vice versa are treated as inner junctions/symbolic links, because all selection is treated as a common root.

Smart Move Smart Move enables folders with junctions and symbolic links beneath to be renamed, and the junctions and symbolic links' targets are updated below that folder. Without Smart Move renaming of such folders would end in dead junctions and symbolic links.

With junctions or symbolic link directories it behaves as follows:



By closely looking at the above picture one can find three different types of folders/junctions:

Normal Folders The folder 'Folder 3' is a normal folder. It gets moved with its content straight forward.

Inner Junctions Symlinks The folder 'Inner Junction/Symlink' is targeted at 'Inner Junction/Symlink Target'. In LSE naming universe this kind of folder is called *Inner Junction/Symlink*, because its target points to a folder, which is below the common anchor 'Folder 1'.

Inner Junctions/Symlink are updated properly via Smart Move in the destination location.

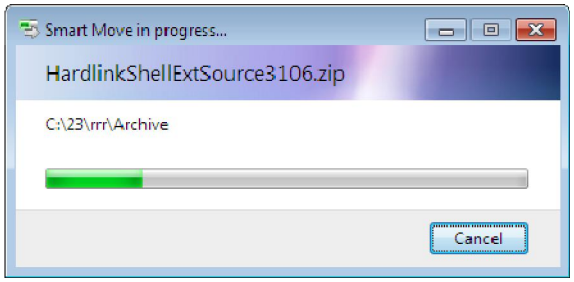
Outer Junctions Symlinks The folder 'Outer Junction/Symlink' is targeted at the folder 'Outer Junction/Symlink Target'. In LSE naming universe this kind of folder is called *Outer Junction/Symlink*, because its target points to a folder, which is in paralell and thus outside the anchor 'Folder 1'.

Outer Junctions/Symlinks are not touched by Smart Move and thus stay connected to their respective target. Please note that this is different to Smart Copy, which has 3 different ways to deal with Outer Junctions/Symbolic Links.

The Smart Move functionality is integrated into Explorer seamlessly, so that you don't have to do anything special. Simply drag a folder in explorer to its destination location, or e.g. press F2 in Explorer to rename a directory and LSE will intercept this operation under the hood,

takes care of junctions or symbolic links, and will update them.

Intercepting move and rename operation means, that LSE takes over control before rename/move, and recursivley searches the selected folder for junctions or symbolic links. But searching large amounts of files and folders takes time, so LSE will show a progress bar when searching takes longer than 250msec.



With Vista & Windows7/8 LSE calls its UAC helper symlink.exe if it has to update symbolic links, so don't be afraid if you get a UAC prompt during moving of folders.

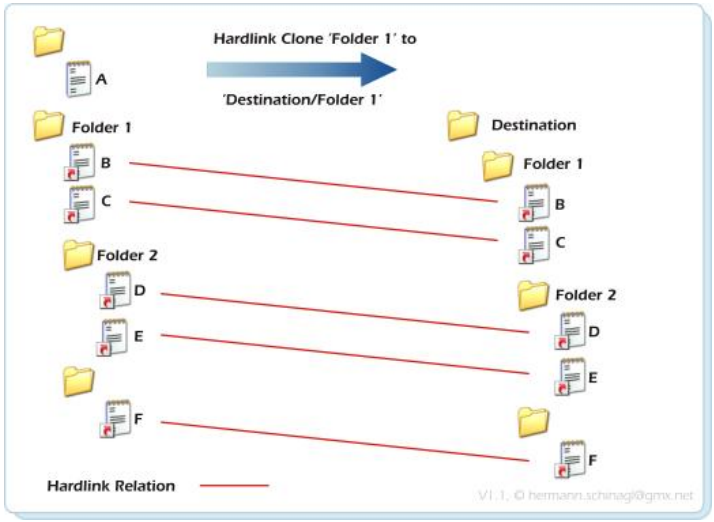
Smart Move creates relative symbolic links during the Smart Move operation.

Enabling/Disabling Smart Move
Smart Move can be switched on/off via the configuration tool

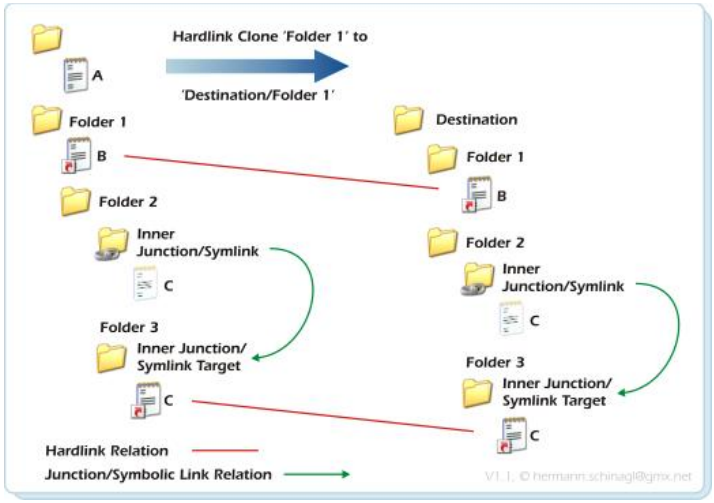
Command Line
The Smart Move functionality is also available via command line from ln.exe via the --move command line switch.

Clone

Clones are copies of a folder tree from a source location recreated at the destination location, however the files within the new folder tree are Hardlinks or Symbolic Links to the respective files in the source folder tree.

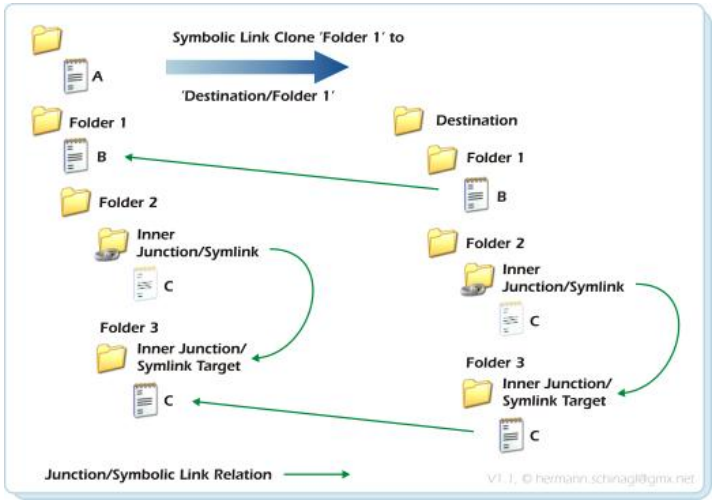


A folder tree might also contain Junctions or Symbolic Links. The clone process recreates inner junction/symbolic links at the destination location very similar as Smart Copy does.

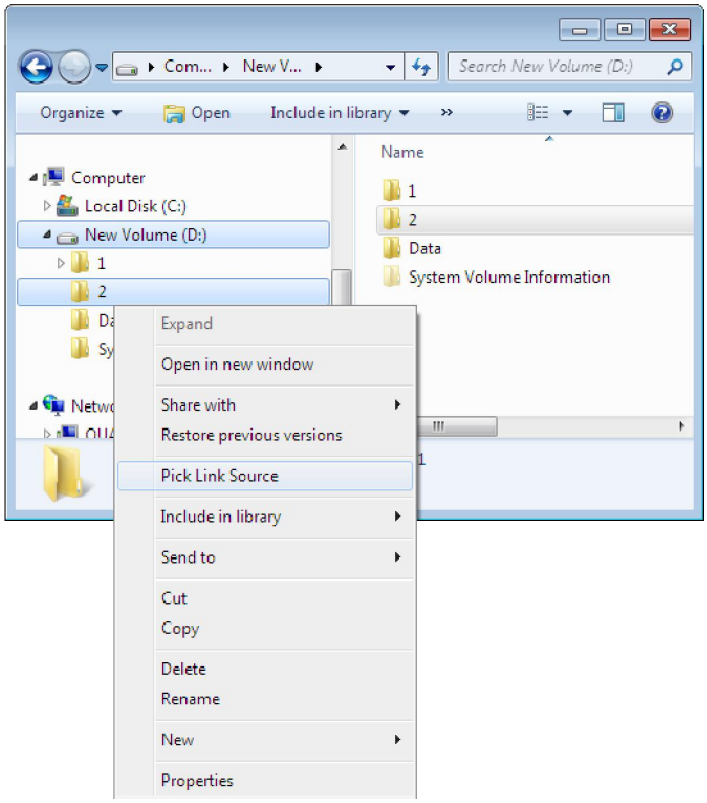


Outer Junctions/Symbolic links are recreated with respect to the specified Outer Junction/Symbolic Link handling. e.g.

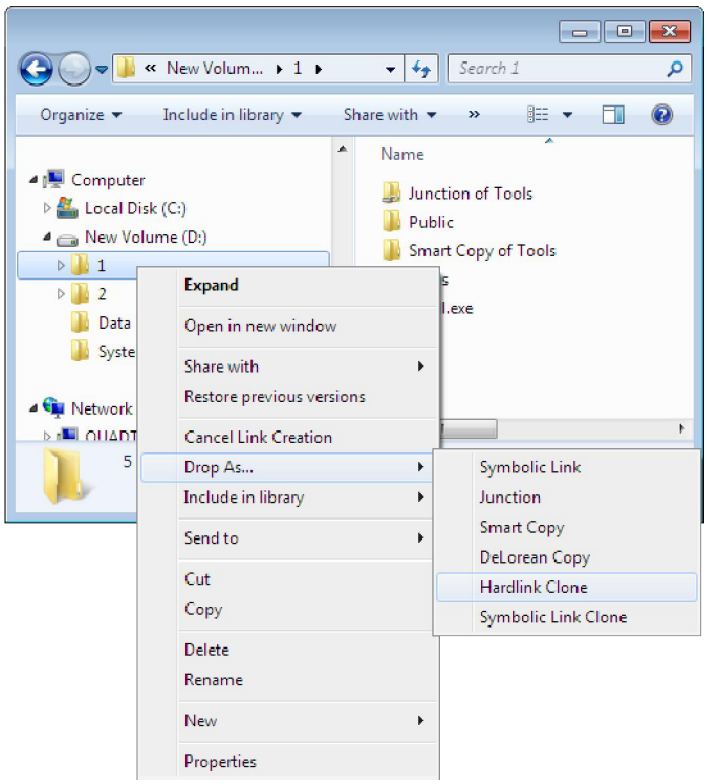
With Windows Vista and Windows7/8 this cloning process is also available with Symbolic Links instead of Hardlinks.



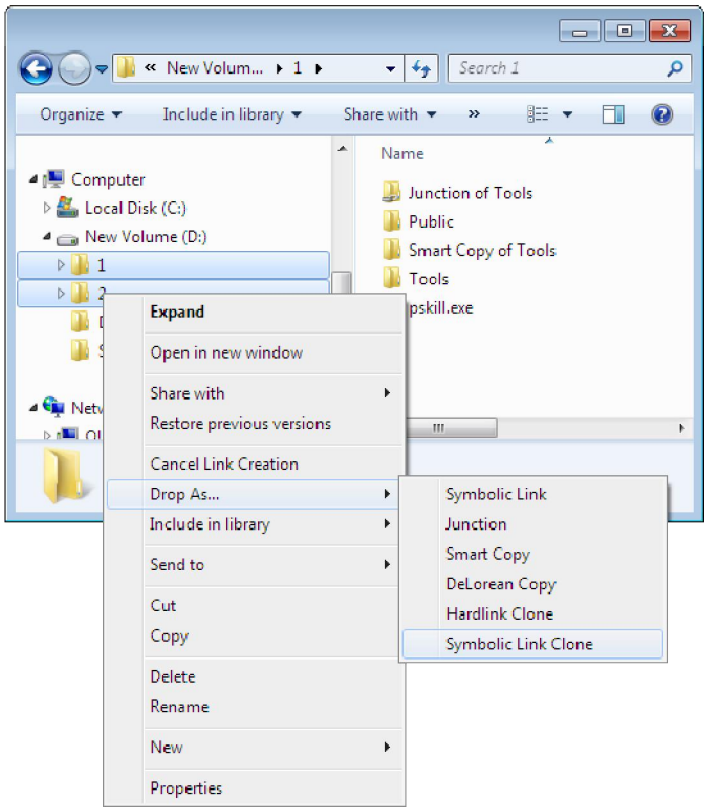
Clones are created in the same way as e.g Junctions. Select a folder, click the Action button, choose **Pick Link Source** from the action menu...



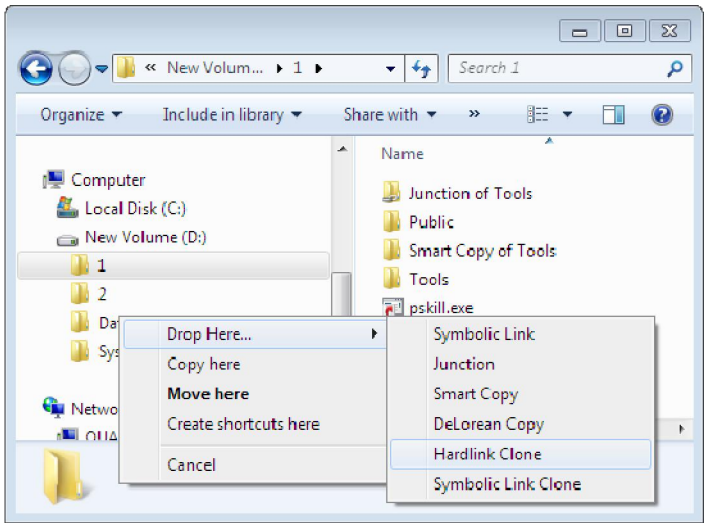
...navigate to the destination folder, press the action button, open the **Drop As ...** submenu and select **HardLink Clone**:



... or with Windows Vista or Windows7/8 choose Symbolic Link Clone to create clones of existing tree structures.



HardLink and Symbolic Link 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** or with Windows Vista and Windows7/8 **Symbolic Link Clone**:



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

If both files 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 or Symbolic Links, they are only available within an NTFS volume. Hardlink Cloning can not replicate the folder structure from one disk volume to a different volume, because Hardlinks are limited to operation on a single volume. Symbolic Link Clones can be used to create volume spanning Clones, but only on Vista or Windows7/8

When creating Clones under Windows7/8 and Windows Vista, LSE forks its helper symlink.exe to forwards this operation to it, if the folder tree contains symbolic links, because the creation of symbolic links needs elevation, and thus brings up the famous UAC dialog.
LSE **only** issues its helper symlink.exe if a symlink is among/below the selected folders, so it saves you from one UAC prompt if you don't

have symlinks among your selection.

Command Line
The Clone functionality via Hardlinks or Symbolic Links is also available via command line from ln.exe via the --recursive command line switch.

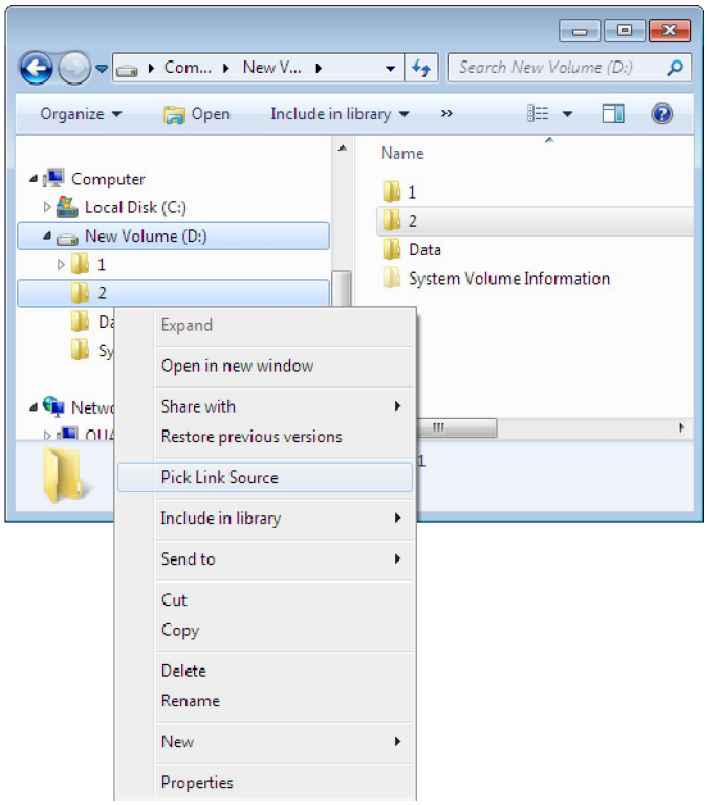
Smart Mirror

Smart Mirror is very similar to Smart Copy and not only copies but *synchronises* the folder *Source* to *Destination*:

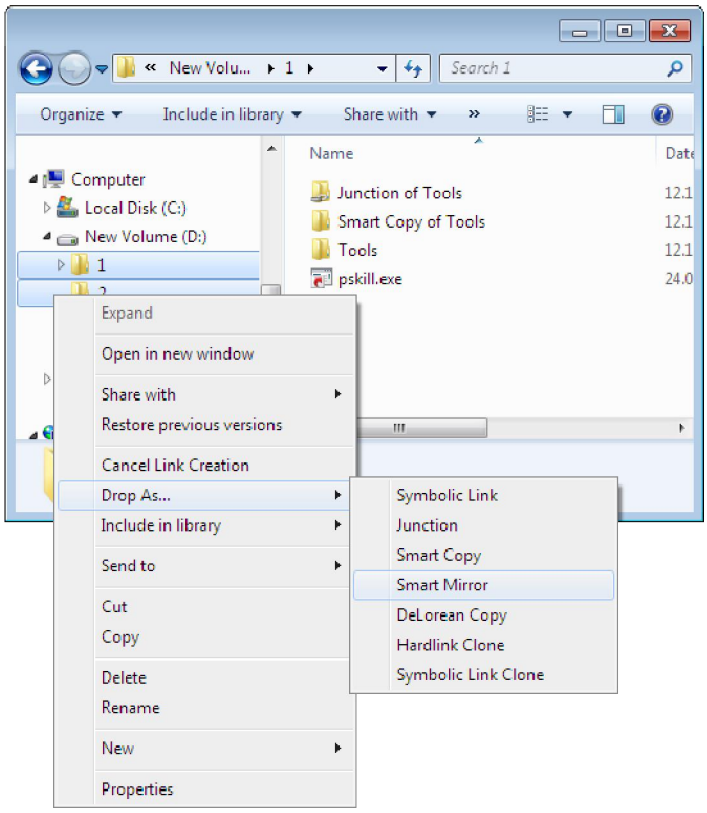
- Smart Copy only newer files from *Source* to *Destination*.
- Delete files not anymore in *Source* from *Destination*.

Smart Mirror takes care of Hardlink Relations, Restores Inner Junctions or Inner Symbolic links or when issued unrolls or splices Outer Junctions or Outer Symbolic Links.

Smart Mirror is created in the same way as e.g Junctions. Select a folder, click the Action button, choose **Pick Link Source** from the action menu...



...navigate to the destination folder, press the action button, open the **Drop As ...** submenu and select **Smart Mirror**:

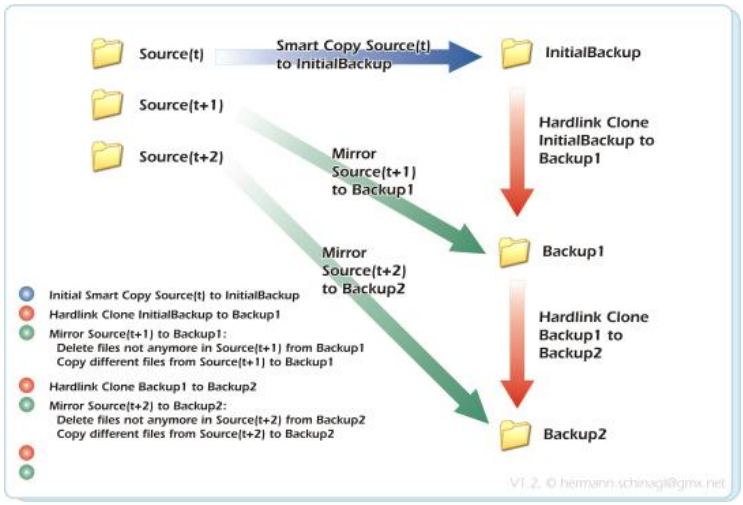


Smart Mirror is a little bit different with respect to Auto Rename, because it expects a already existing folder in the destination location, which has the same name as the source folder, so that it can do the mirror.

DeLorean Copy

DeLorean Copy is a way of creating incremental backups by using a combination of hardlink clone and Smart Copy.

The following picture gives an overview what DeLorean Copy is about



In general a DeLorean Copy has 3 principals: Source(t), InitialBackup and Backup(n).

Phase 1: Intial SmartCopy The folder *Source(t)* is initially copied to *InitialBackup*. This is shown by the blue arrow.

Changes happen During this phase the files under source change, and *Source(t)* becomes *Source(t+1)*.

Phase 2: Hardlink Clone The folder *InitialBackup* is Hardlink Cloned to *Backup1,* which ties *InitialBackup* and *Backup1*. This is shown by the red arrow.

Phase 3: Mirror Mirror the folder *Source* to *Backup1*. This is shown by the green arrow:

- Keep unchanged files as hardllinks to InitialBackup.
- Delete files not anymore in Source(t+1) from Backup1.
- Copy different files from *Source(t+1)* to *Backup1*.

With completion of this first round *Backup1* contains the first lean and mean copy of *Source* only consisting of either hardlinks to *InitialBackup*, or of copied files if there was the need to copy them over from *Source(t+1)* because they were newer under *Source(t+1)*.

The point is that all files in *Backup1* are transparently accessible, but really little space is used, because not all files in the *Source(t+1)* changed, so that there was only the need to effectively copy over a few files from *Source(t+1)* to *Backup1*.

This can be repeated on and on. The second round would be using *Source*, *Backup1* and *Backup2* for DeLorean Copy:

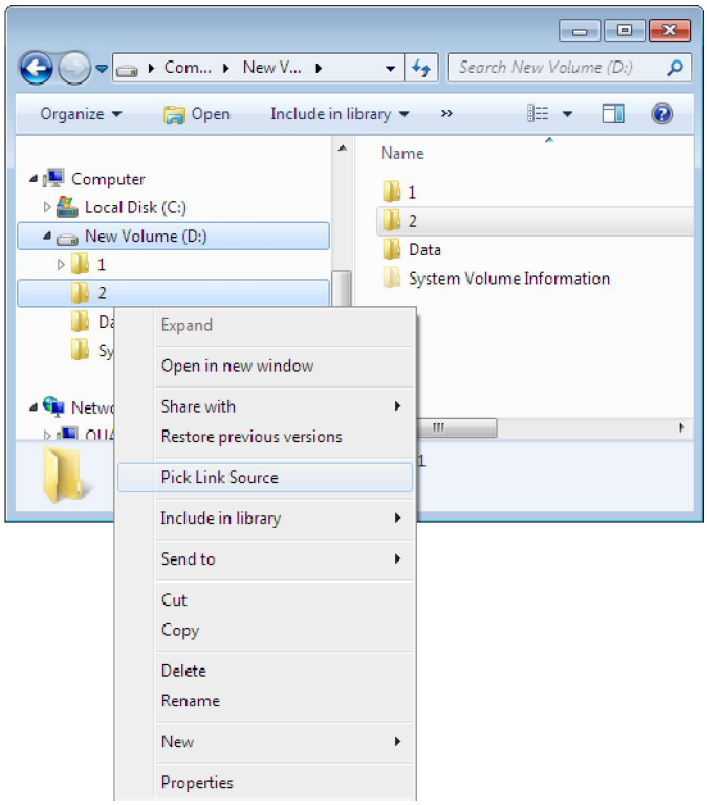
Changes happen During this phase the files under source change, and *Source(t+1)* becomes *Source(t+2)*.

Phase 2: Hardlink Clone The folder *Backup1* is Hardlink Cloned to *Backup2*, which ties *Backup1* and *Backup2*. This is shown by the red arrow.

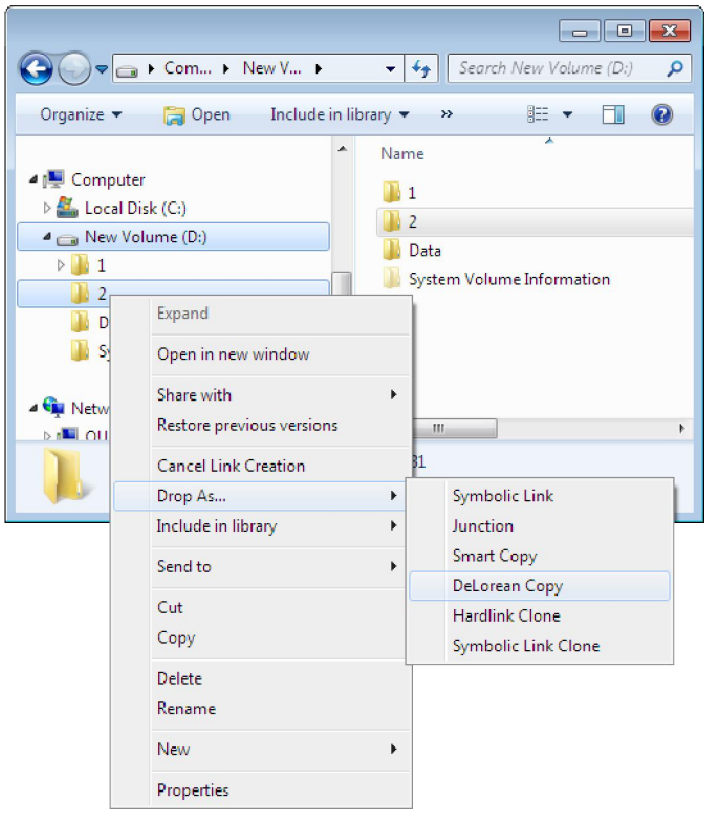
Phase 3: Mirror Mirror the folder *Source(t+2)* to *Backup2*. This is shown by the green arrow:

- Keep unchanged files as hardllinks to Backup1.
- Delete files not anymore in Source(t+2) from Backup2.
- Copy newer files from *Source(t+2)* to *Backup2*.

DeLorean Copies are created in the same way as e.g Junctions. Select a folder, click the Action button, choose **Pick Link Source** from the action menu...



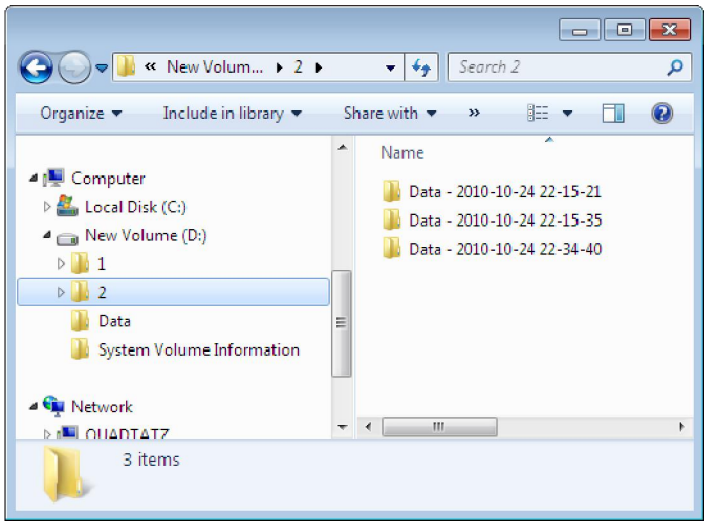
...navigate to the destination folder, press the action button, open the **Drop As...** submenu and select **DeLorean Copy**:



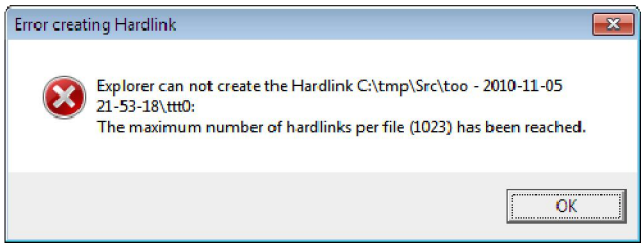
If a DeLorean Copy was dropped the first time onto a directory, the operations, which are described under phase 1 in the above descriptions, namely a Smart Copy, is performed. Link Shell Extension automatically generated the folder name for the destination by appending a timestamp.

Any successive drop of a directory onto the a destination directory initates Phase 2 and Phase 3 from the above description, namley it does the Hardlink Clone from the former backup onto the current backup and furthermore does mirror the source onto the current backup.

A directory holding many copies may look like this.



Limitations
It is little known, but NTFS has a limit to create a maximum of 1023 hardlinks to one file. For DeLorean Copy this means that it will display an error message if this limit is exceeded, because exceeding this limit means loss of data among the most recent backup sets:



The reason for exceeding this limit could either be, that there have been more than 1023 backup sets but no hardlinks within the *Source*, or there are hardlinks within the *Source* and less than 1023 backup sets.

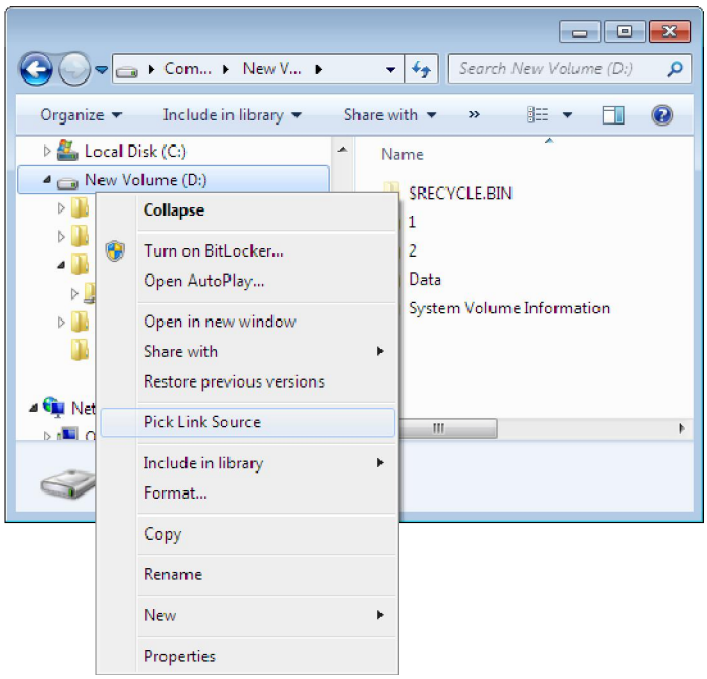
The DeLorean Copy submenue will **not** appear if **more than one** folders are selected as source.

DeLorean Copy is long path safe which means it can handle more than 256 characters in pathes. This is important, because placing a copy with quite long path, but still below 256 characters path length, to destination locations might result in combined path length greater than 256 character. In such a situation no data loss must happen, which Delorean copy guaranties with beeing 'long path safe'. Please make sure that Explorer can not show you the result of such a copy, but the files are there. Alternative explorers like SpeedCommander can handle this.

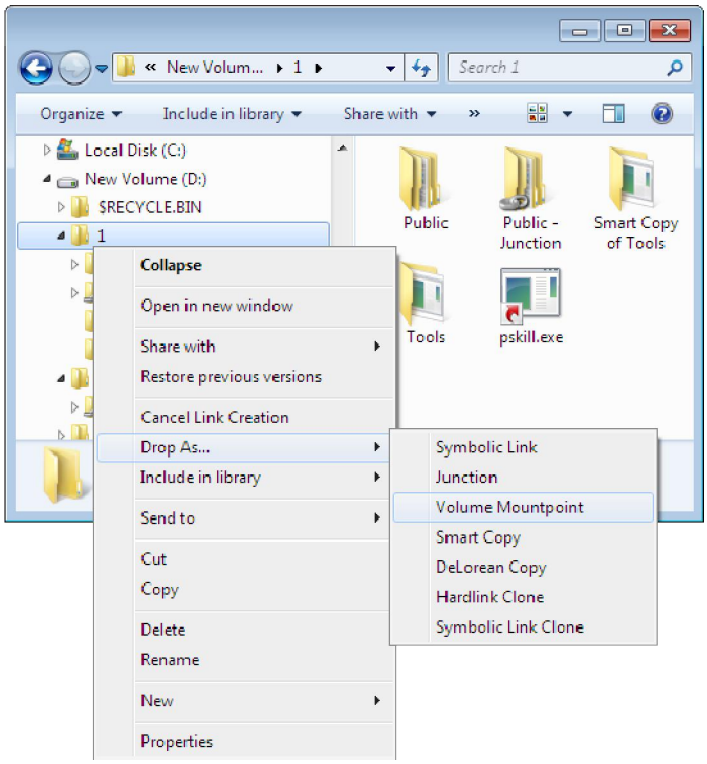
Command Line
DeLorean Copy functionality is also available via ln.exe

Volume Mount Point Support

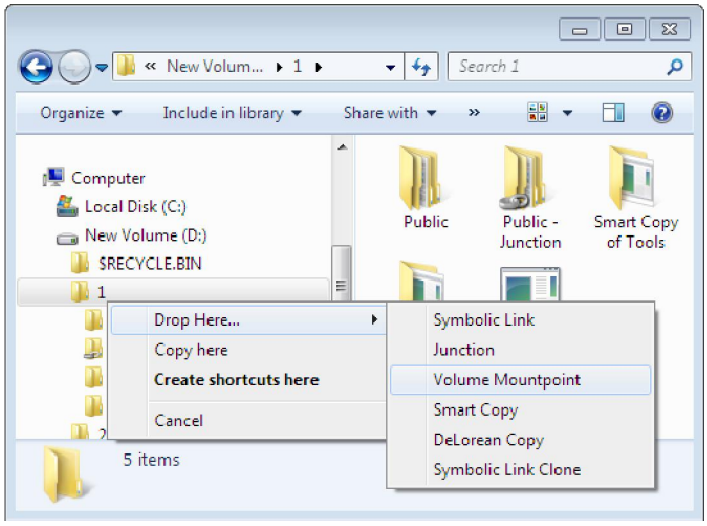
Volume Mountpoints are a feature of NTFS version 5.0, which provides functionality to map complete local volumes onto arbitrary disk locations. Volume Mountpoints were not supported in NTFS Version 4.0



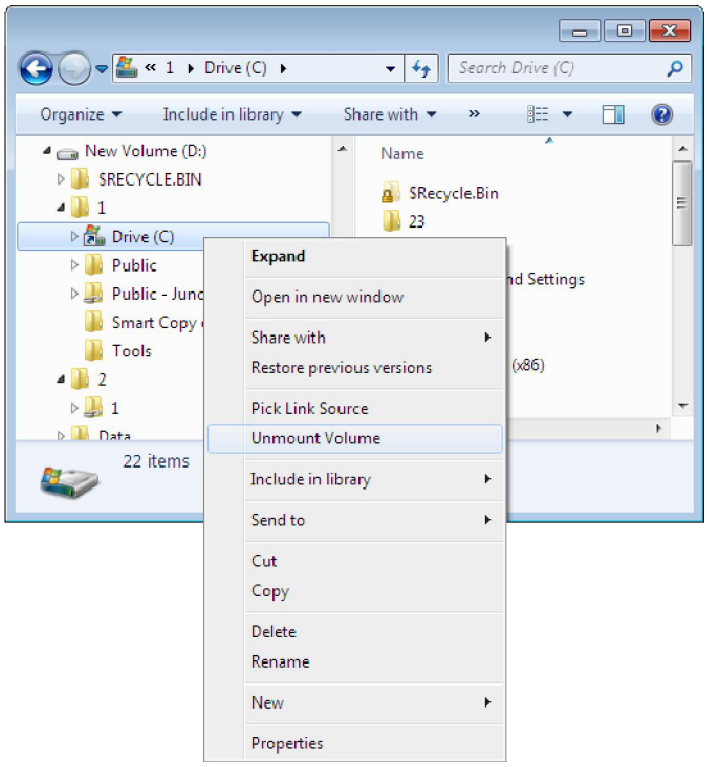
Volume Mountpoints are created in the same way as Hardlinks, except that the Source Link is a volume rather than a file. Select a local volume, click the right mouse button, choose **Pick Link Source** from the action menu, navigate to the destination folder, click the action button, open the submenu **Drop As ...** and select **Volume Mountpoint**:



Volume Mountpoints can also be created via Drag and Drop when the selected local volume is dragged with the action button pressed to a destination folder; when the right mouse button is released, select the **Drop Here ...** submenu and then **Volume Mountpoint.**



Mount Points can be deleted by using the Unmount Volume command from Explorer as usual.



To show the origin of a Volume Mountpoint, the reference column of a Volume MountPoint shows the volume which is mounted onto the selected path.

Make sure that only local volumes can be mounted but not mapped network drives.

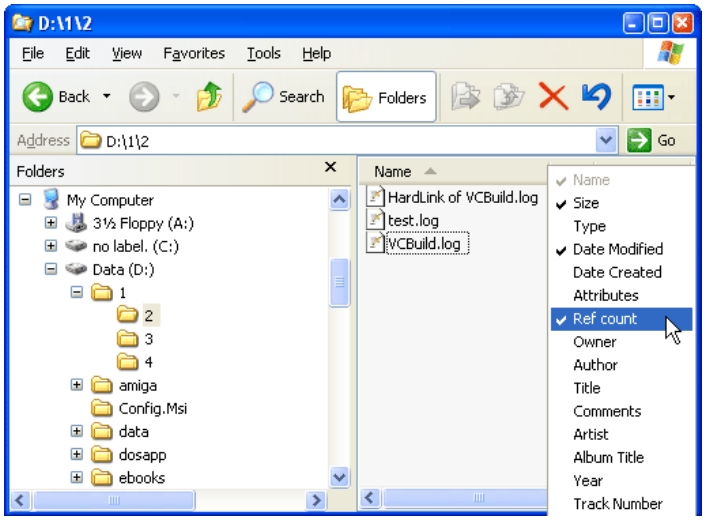
Vista & Windows7/8: With Vista & Windows7/8 the creation and deletion of Volume Mountpoints is bound to successful elevation, which means that the famous UAC dialog must be acknowledged.

@Microsoft: Why is it not possible with Windows7/8 & Vista to create a Volume Mountpoint under e.g. 'c:\Program Files'. The return value on the CreateMountPoint() call is ERROR_INVALID_PARAMETER, which is always the 'Error I don't know what to do' thrown by Windows. Please fix this!

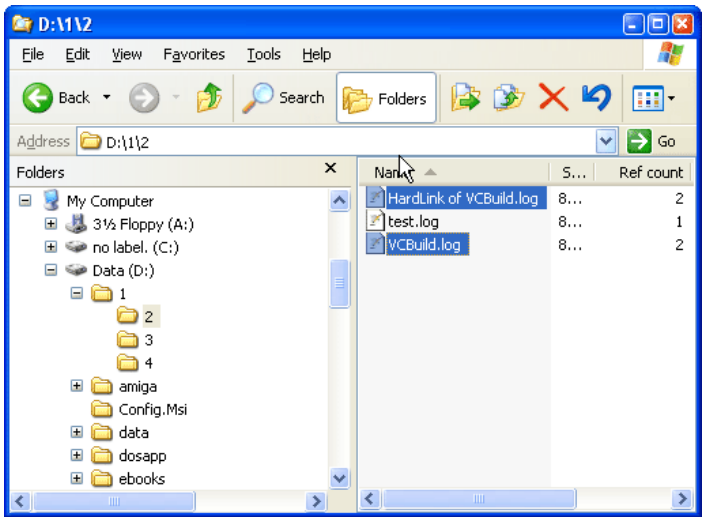
Reference Count

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.

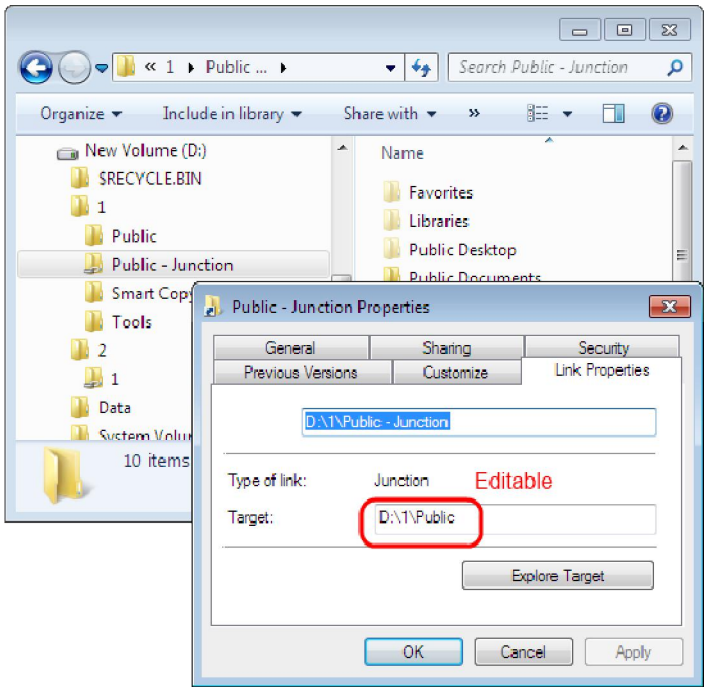


Vista & Windows7/8: In the current release of Link Shell Extension the column, which shows the reference count and the origin of the junction is not available, because the way Vista and Windows7/8 handles user defined columns has been completely revamped by Microsoft and all applications working with so called ColumnHandlers have to be partially rewritten.

Link Properties

Link Shell Extension also supports so called Explorer Property Sheets, which means that if a file or directory property in explorer is opened, Link Shell Extension adds its own tab to show the properties of a hardlink, junction, volume mountpoint or symbolic link.

This additional tab only shows in the file or directory properties, if the file or directory is a hardlink, junction, volume mountpoint or symbolic link, otherwise this tab is not available.



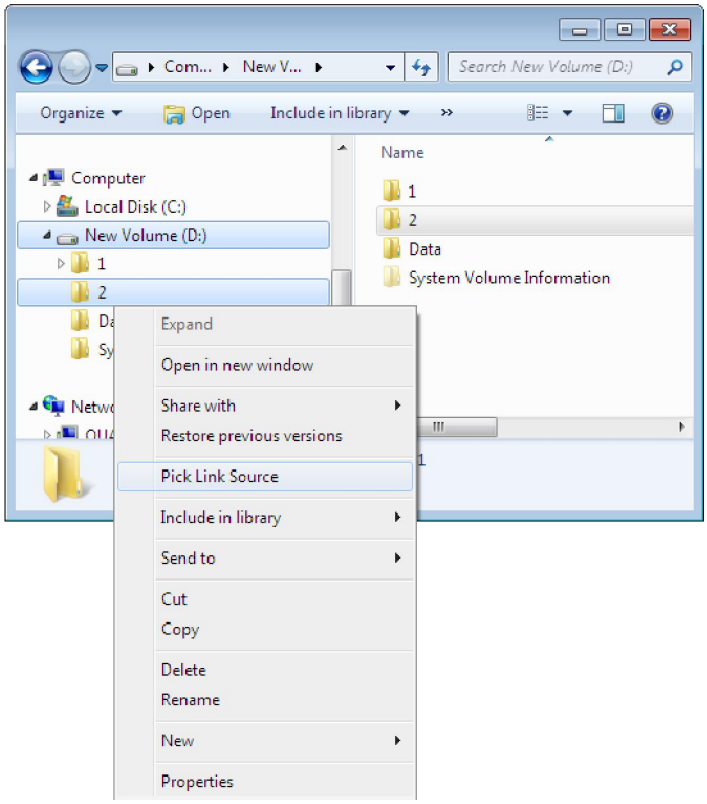
Explore
For junctions, volume mountpoints or symbolic links this dialog also shows a 'Explore Target' button, which opens an explorer in the specified directory.

Edit
For Junctions or Symbolic Links the Target field can be edited, and after either pressing the apply button or leaving the Link Property dialog with ok, the changes are applied to the Junction or Symbolic Link.

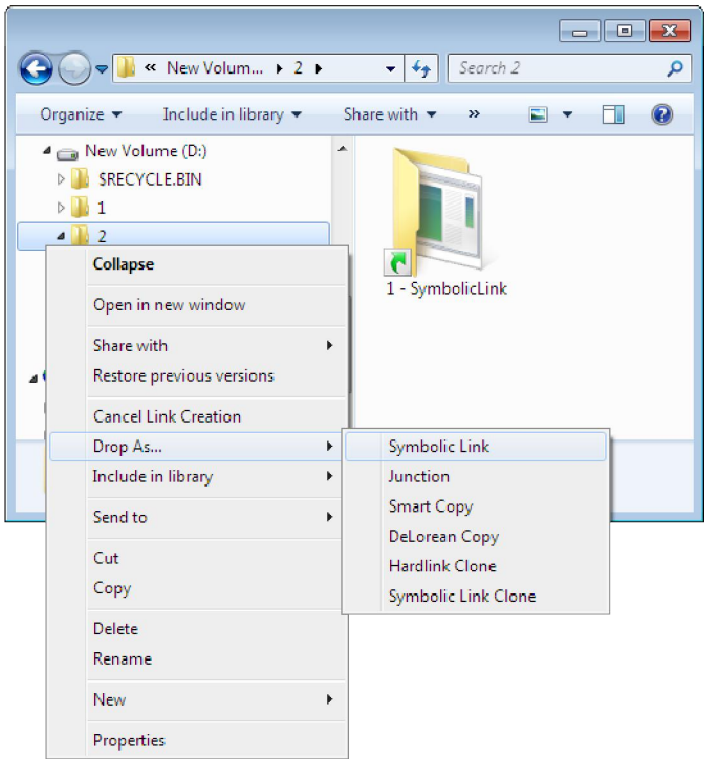
SymbolicLink with Vista and Windows7/8

With Windows Vista and Windows7/8 NTFS introduces a new type of link, the *Symbolic Link*. LSE supports the creation of symbolic links under Windows Vista and Windows7/8.

Creating a **Symbolic Link** is essentially the same as the other Link creation processes. Action click on the selected file(s) and select Pick Link Source(s) from the action menu.



Under Vista/W7 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.



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 newly created *Symbolic Link Clone* folders.

Symbolic Links can also be created between directories.

Relative versus absolute symbolic link target pathnames
The target of a symbolic link can either be

- a fully qualified path starting at the root of a drive, e.g e:\data\cpp\myfile.txt
- or can be be specified relativeley, e.g ..\..\data\cpp\myfile.txt

LSE by default tries to create **relative** target path names for symbolic links as long as this is possible, e.g the file and its target are on the same logical drive. Having relative symbolic link targets is much smarter especially when the target of links is in the same directory. If a symbolic link and its target are on different drives, LSE uses absolute pathnames.

The configuration tool can switch Link Shell Extension in either relative or absolute mode.

Overlay Icons for Symbolic Links
To help distinguish Symbolic Links from normal files/directories, an **overlay icon** is implemented on symbolic links that shows a light green arrow icon under the folder.

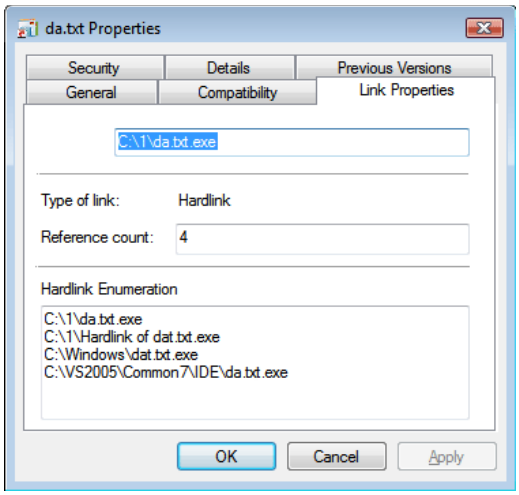
Overlay icons for Symbolic Links can also be customized.

@Microsoft: Symbolic links without elevation: Lets see if this paragraph is read by someone. I don't know if it is a bug or a feature, but one can create symbolic links without elevation, if you create a hardlink from a symbolic link. That sounds weird, but it works. And by a closer look at this stuff, it can be seen, that LSE really uses CreateHardlink() and a symbolic link is created. MS guys! Your opinion?

@Microsoft: Symbolic can span network drives as long as the target is a drive letter mapped drive, but it fails for UNC names in symbolic link target with error ERROR_LOGON_FAILURE(1326), even if the network is connected properly.

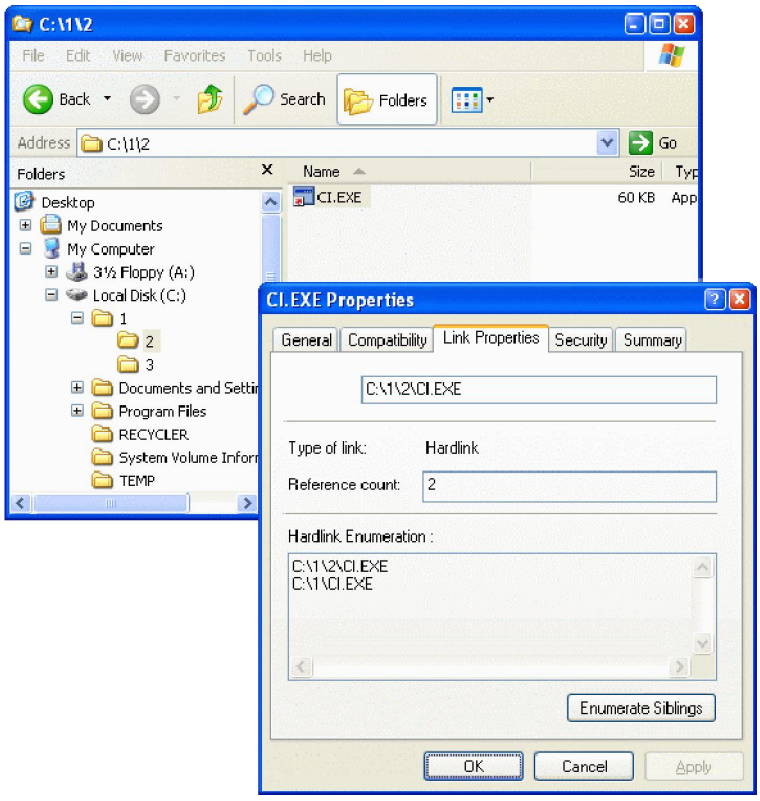
Enumeration of With Vista and Windows7/8 it is possible to enumerate all hardlink siblings to a file in constant time. Simply select a hardlinked file and select

Hardlinks Properties from the right click action menu:



Unfortunately this feature is currently only available with Vista since the Win32 API calls to enumerate hardlinks are not available with XP or W2K.

Under XP reading and processing the unique file ID of all filenames on a disk is necessary to gather the same information. Since this operation is timeconsuming, the siblings of a hardlinked file are not shown immediately after opening the properties tab, but the 'Enumerate Siblings' button must be pressed.

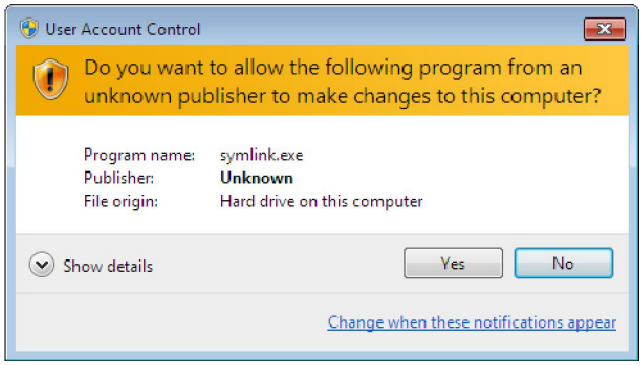


Enumeration of all siblings under XP can take time, because in the worst case all files contained on a logical volume must be at least opened for the filename.

The Hardlink Enumeration functionality is also available via command line from ln.exe via the --enum or the --list command line switch.

UAC One of the major Windows7/8 changes was the so called User Account Control (UAC). Due to UAC some API calls need elevation to administrative level, and this elevation must be acknowledged via the below shown dialog box. Unfortunately, and I still can't believe this, the API call CreateSymbolicLink is a call, which needs elevation, and thus causes this annoying dialog box come up every time a symbolic link is

created.
So if you see the below box, and the program asking for elevation is symlink.exe, it is Link Shell Extensions contribution to UAC and you must acknowledge it to get Symbolic Links created.



UAC will be the reason, that many applications simply either will not work, or are not useable with Windows7/8. They are not useable, because privileged API calls and non privileged API calls are mixed in the code, and first need to be separated by architectural means into an separate .exe, which gets elevated, and an .exe for the rest of the coding.

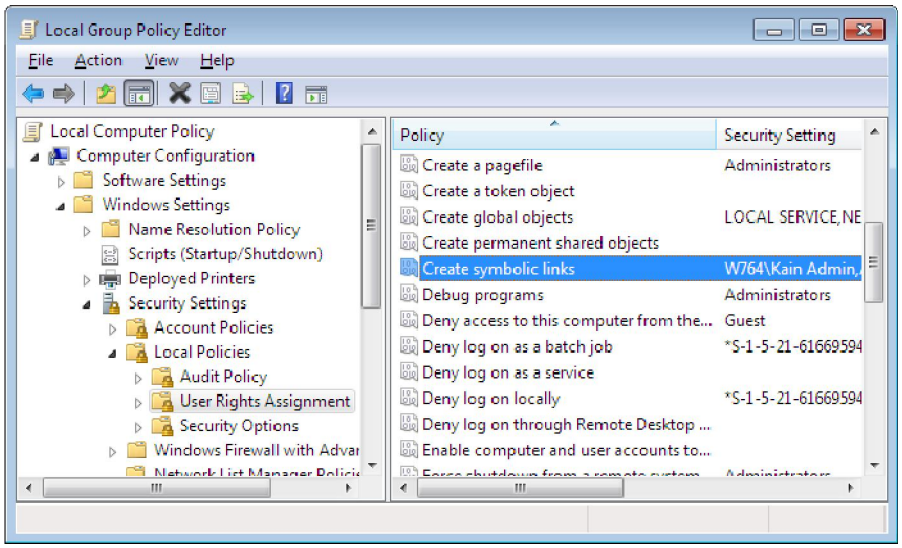
Many applications will simply fail, because calls like CreateProcess(), which are very common, do not work, if the process started with it needs elevation. Well the story goes on and on, and Link Shell Extension is a very small application, but to walk the learning curve was a hard way, especially because the documentation was rather difficult to get.

I guess the main protection of UAC is that many small legacy apps are impossible to rewrite for Windows7/8, and they simply can not be used with Windows7/8efficiently.

Change Symbolic Link Privilege

One way to come around the UAC prompt for the creation of Symbolic Links is to globally allow users to create Symbolic Links by changing the policy:

For specific users the permissions can be granted/revoked via gpedit.msc, under "Computer Configuration" ->"Windows settings? -> "Security Settings? -> "Local Policies? -> "User Rights Assignments?, and is called "Create Symbolic Links?.



Linkshell Extension can deal with the above granting of privileges, and if the Symblic Link Privilege is available avoids the UAC prompt.

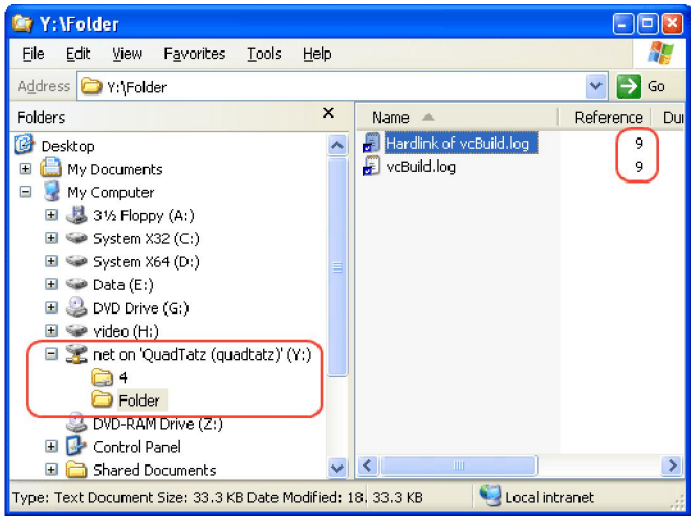
Remote Capabilities

It is little known, but the SMB networking protocol supports operations to create remote Hardlinks, Junctions and Symbolic Links within SMB mapped network NTFS drives.

This feature is used by Link Shell Extension, to enable the creation of so called Remote Hardlinks, Remote Junctions, or Remote Symbolic Links. e.g.

- Map a network share
- Pick a file from that share
- Drop the file as Hardlink within the same share

A Hardlink has been created, which can be easily verified



Furthermore SMB1.0 also reports the reference count for Hardlinks and the junction origin for Junctions, which enables Link Shell Extension to show the properties dialog for remote files. Currently the reference count of a hardlink is reported via SMB1.0 in 90% correctly, so please be aware of this restriction.

@Microsoft: Why does this happen in 90%? I have scanned my code in the meantime many times, and it is obvious, that the same call sometimes succeeds and sometimes not. Why??? If you wanna hunt down a bug in your code, please contact me!

LSE supports both, mapped network drives and UNC paths.

Mapped but not available network drives can in general be the reason for sloppy explorer startup performance. Delays of a few seconds can be experienced if explorer has to check all drive mappings, especially the ones which are not available. This gets worse, if LSE also checks the status of all drives.

To workaround this caveat the Remote Capabilities of Link Shell Extension can be switched on/off via the configuration tool.

Remote Hardlinks and the SMB version
There are differnet SMB Version implemented in different Windows versions which means different behaviour for hardlinks:

SMB1.0: Windows XP, Windows2000 ...
SMB2.0: Windows Vista ...
SMB2.1: Windows7/8, Windows Server 2008 R2 ...

All of those version support the creation of hardlinks remotely, but since SMB2.0 one can not find out if a file on a remote drive is a hardlink or not.

This means, that e.g if you connect with your Windows XP machine to a SMB2.1 drive, which is provided by a Windows7/8 machine, you will not be able to see overlay icons for hardlinked files, but you will be able to create them remotley.

@Microsoft: Is this implementation since SMB2.0 a bug or a feature?

Removeable Media

LSE supports removable media, which have been formatted with NTFS, to create all kind of features it does for fixed drives too. The only limitation is that it intentionally won't work on removable media if they are mounted to drive A: or B:. The reason is that A: or B: are commonly used for floppy drives.

With removable media formatted to NTFS there is the slight chance that LSE reports 'Access denied' problems, when creating hardlinks or junctions. This is due to file object permissions on the removable NTFS drive, which have been created with a different computer on that removable media, thus causing this 'Access denied' messages. The solution here is to change the permission on that removable media as Administrator.

Very long Path

The Win32 API supports pathnames up to 256 characters, thus limiting all applications to that length for pathnames.

On the other hand NTFS supports pathnames with up to 32767 characters, so one might have already experienced pathnames, which are longer than 256 characters. To deal with that, LSE can handle *Very Long Path* up to 32767 characters with all operations.

Subst Handling

With the subst.exe command one can create driveletters, which point to certain path on a NTFS volume. This means that two different driveletters in the end might resolve to locations on the same NTFS volume.
Link Shell Extension checks these situations when it comes to allow the creation of hardlinks, and as a consequence allows the creation of hardlinks among different logical drives if they resolve to the same NTFS volume.

Symbolic links for Windows XP

Technically it seems that NTFS5 even shipped with WindowsXP has always supported symbolic links, but the functionality was not available for user mode applications, and even not for the upper layers of ntfs.sys

BUT: There are filter drivers available from Masatoshi Kimura for even WindowsXP, which enable symbolic links under XP. These drivers can be downloaded from his home page for 64 bit and for 32 bit. The sources are also available.

Installation of these drivers is also quite simple.

- Unpack the archive for your plattform,
- Open a command shell with administrator access privileges on a directory, containing both, senable.exe and symlink.sys, to run following commands:

  - **senable.exe install** copies the driver symlink.sys to %systemroot%\system32\drivers, registers it at HKLM\SYSTEM\CurrentControlSet\Services\SymLink and starts it.
  - **senable.exe delete** deletes the driver symlink.sys from %systemroot%\system32\drivers and registry, but nothing more. Once copied to %systemroot%\system32\drivers and registered, it still needs to be activated by seneable.exe on each Windows start. To unload the driver after deletion you need to reboot Windows, logoff is not enough.

To start the symbolic link driver automatically, change the registry value HKLM\SYSTEM\CurrentControlSet\Services\SymLink\Start from 2 (automatic) to 0 (boot). From now on, symlinks automatically become working after each Windows start, even if directly logged into your limited account.

Link Shell Extension enables its complete symbolic link functionality under XP as soon as it finds this driver loaded. And, sensation(!): There is no UAC prompt necessary under XP for creating symbolic links ;-))

Furthermore there are reports from users who tried out the driver and LSE also under Windows2000, and everything works fine.

ReFs Support

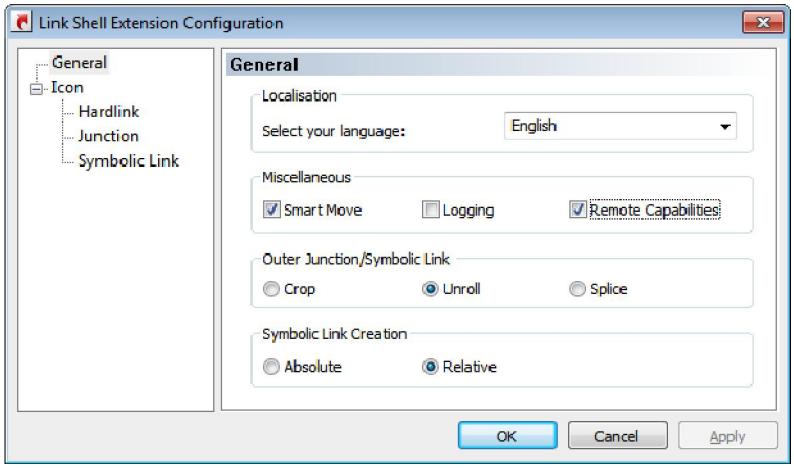
With Windows Server 2012 Microsoft introduced the ReFS filesystem, which is the designated successor to NTFS. But the first implementation of ReFS can do nice things, but lacks a few important features from NTFS like the Hardlink support.
Link Shell Extension supports ReFS so that one can create Symbolic Links, Mountpoints, Junctions on ReFS volumes. But it will throw an error message if a hardlink is about to be created on a ReFS volume. So what are the impacts on Link Shell Extension in detail:

- ReFS drives as the destination of Delorean Copies will for sure fail, at least when creating the second backup in a Delorean set.
- SmartCopy/SmartMirror will fail as destination for hardlinks within the source.
- SmartMove will work.

Since ReFS is expected to support the full set of NTFS functionality in a later release, Link Shell Extension has no checks implemented to tackle with the above constraints.

Configuration

Link Shell Extension can be tweaked/configured to fit the different personal taste in some aspects. To ease this, Link Shell Extension has a companion called LSEConfig, which changes Link Shell Extension behaviour via a User Interface. Once started, LSEConfig will throw the famous UAC UAC dialog, because Link Shell Extensions settings are changed in the Windows registry.



Localisation
Link Shell Extension's UI and commands are available in a few languages. You can choose from

- English(default)
- Chinese
- Czech
- French
- German
- Italian
- Japanese
- Polish
- Portuguese Brazilian
- Russian

- Slovak
- Spanish
- Swedish
- Turkish

Changing the UI Language of Link Shell Extension will need a restart of the explorer once Apply or Ok is pressed.

Smart Move
It might be usefull to totally switch off Smart Move, if there are folders with really much folders. This can be acchieved by ticking the *Smart Move* checkbox.

Logging
All output of a LSE operation like SmartCopy, SmartMirror, or Delorean Copy is logged to the file %TEMP%\LinkShellExtension.log

Remote Capabilities
It might be usefull to totally switch off Remote Capabilities, if there are lots of 'dead network drives' around. This can be acchieved by ticking the *Remote Capabilities* checkbox.

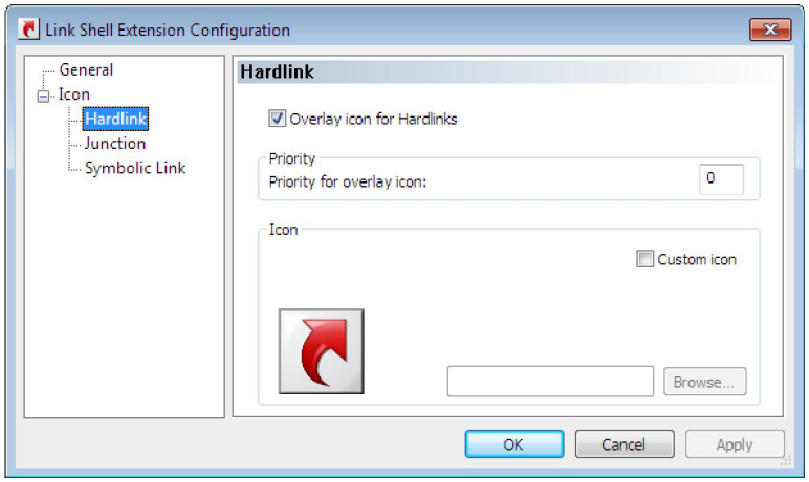
Outer Junction/Symbolic Link Handling
Decide whether Outer Junctions should be handled as *Crop*, be *Unrolled*, which is the default, or *Spliced*.

Symbolic Link Creation
By selecting either *relative* or *absolute* Link Shell Extension will create the target of Symbolic Links respective.

Custom Overlay Icons

Link Shell Extension has built in overlay icons for junctions, hardlinks and symbolic links. Since icons are subject to individual taste, the icon used by Link Shell Extension can be specified.

Changing any settings related to overlay icons will ask for a restart of explorer when Apply or Ok is pressed.



Overlay Icon
Sometimes it might be usefull to totally disable certain overlay icons from Link Shell Extension, which can be acchieved by ticking the checkbox for *overlay icons*.

Priority
Only one overlay icon can be shown with an icon, but many overlay handlers might apply to provide the overlay icon. To sort this out each overlay handler can specify a priority to explorer and explorer shows the overlay icon with highest priority.
High priority means lower number, with 0 as the highest priority

Custom Icon
By ticking the checkbox for *custom icon* the *Browse...* button gets enabled, and an icon can be selected. Keep in mind that custom icons are specific to each user.

General
Windows Vista and Windows7/8 are a little bit special, because overlay icons for 256x256 must not be in the lower left corner of the icon, and must not be already smaller to perfectly 'overlay' an icon. 256x256 overlay icons must fill up the complete available icon, and also must not be resized. Vista does that for you for 256x256 icons.

Or in other words Vista takes any 256x256 icon and resizes it to 92x92, moves it to the left lower corner and overlays.

For all other resolutions smaller than 256x256, Vista works in the same way as XP, you have to prepare an overlay icon in the lower left corner.

Furthermore Vista icons should not be saved 'compressed', because XP can not read compressed icons.

For my investigations the icon editor of choice capable of dealing with Vita icons was RealWorld Icon Editor

Apply Changes
When you press OK or Apply on LSEConfigs dialog, settings will be taken over.

If changes were made to Link Shell Extensions language settings or settings related to overlay icons, you will be asked to confirm a restart of explorer.exe, so that your changes become effective. Restarting explorer.exe means, that e.g. any copy operation or other pending operation within explorer.exe is interrupted.

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.

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 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.

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/W2K3/W2K3R2/W2K8/W2K8R2/WXP64/Vista/Vista/Windows7/Windows8 in 32bit, 64bit or Itanium.
- Hardlinks can only be made on NTFS volumes, under the supported platforms.
- Hardlinks can only be made within one NTFS volumes, and can not span across NTFS volumes.
- Junctions can not be created on NTFS volumes with NT4.
- The *Pick Link Source* 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 *Pick Link Source* 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.
- There is an OS limit of creating more than 1023 hardlinks per file. This is less known, but it is there.
- ReFs does not support hardlinks.
- Windows XP does by default not support symbolic links, but there is a driver to enable symbolic links also under Windows XP

Frequently Asked Questions

- **Q:On Windows7/8, the Save As... box shows symlinks with the white "shortcut" overlay, instead of the green symlink overlay.**

  A: This happens if the processes shown during installation of Link Shell Extension were not closed. If you really run into this rare situation, a reboot will help.

- **Q:However the value of the reference count is not updated when hardlinks are deleted. That is, when I add new hardlinks the value increases properly, but when I delete hardlinks, the value does not change. Is that a bug? Or there is a way of refreshing the Windows Explorer?**

  A: Once a file is deleted in Explorer it is moved into Recycle Bin, but not really deleted. If you press Shift-Del for deleting a file instead of just pressing Del, the file really gets deleted and the reference count is decremented.

- **Q:I could'nt make a successful hardlink for image or vector files - I mean, I was able to *make* the hardlink copy, but when I modified one file it didnt affect the other. I'm wondering do you know why this might be - could it be my otherwise quite normal computer (!) or could it be something to do with the hard link proccess ?**

  A: You were able to make hardlinks successfully, but when you open a hardlinked file for **edit**, it depends on the editor associated to the file if the file either gets

  - opened, changed, the original deleted, and the new one saved (==> link broken)
  - opened, changed, and saved back (==> link alive)

- **Q:When I deleted a directory, its junction point is left behind in an non-operational state.**
  Is there a way to prevent this? That is, for example, is it possible to automatically delete the junction points if the associated source is deleted? Or, is it possible to have a program prune such orphaned junctions afterwards?

  A: No sorry, Junctions are a one way relation, and if the targets disappears the junction points to an orphaned destination.

  If you have SmartMove enabled, at least inner junctions/symbolic links are adapted

- **Q:When I delete a symbolic link, which points to a zipped folder by pressing DEL, later on when I want to empty the recycle bin, explorer denies by showing me error message 0x80071128. What's wrong?**

  A: Unfortunatley this is a bug in Explorer, and it only happens to symbolic links pointing to .zip files. The workaround is to move it manually out of recycle bin rename it, and then delete it once more.

- **Q:I have created a symbolic link to an .exe and when I double click on it, I get the following error message:**
  The specified path does not exist. Check the path and try again.

  A: Unfortunatley this is a bug in Explorer, and I don't have a clue how to come around this in explorer.
  If you start the symlink to an .exe from a command prompt it works fine, and even third party explorers like SpeedCommander can do this, but explorer seems to have a limitation
  Does anybody know the registry hack to enable this in explorer.exe? Drop me a line.

- **Q:The overlay icons do not show up**

  A: The number of different icon overlay handlers that the system can support is limited by the amount of space available for icon overlays in the system image list. There are currently fifteen slots allotted for icon overlays, some of which are reserved by the system.

  All is controlled by the alphabetical order of OverlayHandlers under

  ```
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ShellIconOverlayIdentifiers
  ```

  If the OverlayHandlers for LinkShellExtension somehow slipped to a slot greater 15 under 32bit Windows or greater 11 with 64bit Windows, the LSE Overlay Icons won't show up.

  To manually boost the priority of LSE OverlayIcons open the above registry location with regedit and rename

  ```
  IconOverlayHardLink --> 0IconOverlayHardLink
  IconOverlaySymbolicLink --> 0IconOverlaySymbolicLink
  ```

  and either restart the explorer or log-off and log-on again. The point here is to change the alphabetical order by prepeding a 0

- **Q:I?m trying to store Dropbox files only on removable storage instead of the internal 128gb of storage. My DropBox already contains lots of files. How do I accomplish redirecting the dropbox folder to the removeable storage?**

  A:
  - Copy the whole dropbox folder from the internal storage under c:\users\[username]\dropbox to e.g x:\data\dropbox
  - Rename the dropbox folder c:\users\[username]\dropbox to e.g c:\users\[username]\dropbox_org
  - Pick Link Source x:\data\dropbox
  - In c:\users\[username] use *Drop as* and select *Symbolic Link* for Windows7 or *Junction* for WindowsXP
  - If everything went fine finally delete c:\users\[username]\dropbox_org

History

in progress — Version 3.8.0.0

- Complete support for UNC path in any Smart Move/Copy/Clone & Delorean Operation.
- Lots of little fixes/improvements, I always wanted to do, but never had the time to.

August 23, 2013 — Version 3.7.5.1

- Dead Junctions to a different drive could lead to not detecting hardlinks during all operations. Very Nasty, but no dataloss caused.

August 4th, 2013 — Version 3.7.5.0

- For Junctions or Symbolic Links the Target field can be edited in the Properties Dialog.
- LSE elevates the creation of hardlinks in system protected directories, e.g.: %systemroot%.
- LSE now offers all its function also in the *Library* folder.
- If enabled LSE summarizes the ouptut of operations in a log file.
- Fixed a crash related to UNC path and Overlay Icons.
- The progress bar showed wrong/incomplete filename-path combinations during operations on large files.
- Symbolic links to mapped network drives can be created now via LSE.
- Replacement functions can be used to repair broken junctions/symbolic links/mountpoints.
- During SmartCopy/SmartMirror/DeloreanCopy the type (absolute/relative) of symbolic link relation is kept in the destination.
- During SmartCopy/SmartMirror/DeloreanCopy the Compression Attribute is copied too.
- The check for prerequisites during install is more accurate aka takes mfc80.dll into account.

- During un-install it is also checked if the hardlinkshellext.dll is held by some process.
- Fixed a problem during SmartMirror when a directory changed into a file or vice versa from one mirror to the next and had exactly the same name.
- The OS Version detection during installation went wrong on certain machines causing symlinke.exe missing from the installation.
- The eunmeration of hardlink siblings didn't work under XP, when the root dir of a drive had to be traversed.
- Tested with Windows8, and thus updated the documentation.
- The x64 version now contains also a 32bit version in one unified install.
- Deinstallation left over a few registry keys.
- During installation not all processes were detected, which blocked the installation.
- Enabled Link Shell Extension on ReFs volumes.
- Added a Swedish localisation. Thanks to Mikael Grönholm.
- Added a Turkish localisation. Thanks to Memet.
- Added a Czech localisation. Thanks to Ashus
- Added a Slovak localisation. Thanks to RobertS
- LSEConfig has been localised.
- The handling of the compression bit during copying/mirroring/deloreaning for files and directories was broken.
- Draging from or Droping to zipped folders caused an explorer crash.
- LSEConfig localized to French.
- Columnprovider shows now shrinked path for junctions if the path is longer than 32characters.
- On some machines LSEConfig always showed up with French. Introduced with 3.749

June 24th, 2012 Version 3.7.2.0

- When working on mapped network drives via SMB or CFIS, as many NAS boxes do, LSE uses a more traditional enumeration mode and this will copy files (which it did not in any case).
- Multiple locations can be selected and the location are treated as a common root with respect to hardlinks/junctions/symbolic links.
- Nested junctions and symbolic links (aka junctions on junction on junctions ...) are now properly restored in any situation.
- Smartmove had problems with relative symbolic links in rare situations.
- Italian translation updated.
- Support for symbolic links under Window XP.
- Can handle subst.exe created driveletters.
- Overlay Icons for symbolic links under Window XP are available now.
- Fixed a few bugs related to WindowsXP and symbolic links handling.
- Elevation to symlink.exe now happens only if UAC is on, or the elevation is really neccesary.
- Fixed a problem with the creation of absolute symbolic links to directories.
- The installer came up with Chinese as default installation language.
- LSEConfig has an About Box, which shows the version of Link Shellextension.
- Replace Symbolic Link failed when not elevated.
- Replace Symbolic Link always created absolute symbolic links regardless of the settings when not elevated.
- Drop Symbolic Link sometimes did not create absolute links when needed in certain situations.
- The installer now shows the language of the installed OS as default.
- The installer provides more info in Control Panel/Program and Features.
- The target of the Symbolic Liks or Junctions can be edited in the properties dialog.
- Non administrators could not create symbolic links.
- With XP and the symlink driver installed the Delete Junction menue didn't show up.
- The status of the privilege for Symbolic Link Creation is checked, so that UAC can be avoided
- Fixed deployment problems for the Win32bit version.

March 9th 2012 Version 3.6.5.3

- Speed improvements during SmartCopy/SmartMirror/HardlinkClone and Delorean Copy.
- Introduced new Heap Manager Rockall for x64 and x86 builds to gain performance.
- Russian translation updated.
- Installation notifies about already running processes, which would make LSE installation fail, because they have loaded dlls from LSE.
- Fixed issue with Windows 8 installation.
- Symbolic Link Icon Overlays installation in registry was wrong, causing problems with the green arrow for symlinks.

April 17th 2011 Version 3.6.0.4

- With all Smart* functionality, Outer Junctions/Symbolic Links can now be unrolled or spliced.
- Added Smart Mirror.
- Speed improvements to Smart Copy, Smart Move and Delorean Copy.
- The configuration tool does not restart explorer for minor changes to the settings.
- Symlinks creation resulted in absolute symlinks even if creation of syminks was specified as relative, if their common ancestor was a root dir.

November 21st 2010 Version 3.5.0.1

- Introduced DeLorean Copy, which is a way of creating incremental copies using hardlinks.
- Fixed creating Symbolic Links from Symbolic Link files created Symbolic Link directories.

October 3rd 2010 Version 3.4.0.2

- LSE now by default creates relative target path names when creating symbolic links.
- Symbolic Links now have an overlay icon.

- Added a configuration tool for LSE options
- Added a priority level for the indidual overlay icons.
- Added an option to switch off overlay icons totally for each type of overlay.
- The Pick Link Source context menue now also shows up on FAT drives, if the item potentially is the source for a LSE operation.
- Overlay icons can be disabled.
- Documented how the install directory can specified when using silent (un)install.
- The menu hydraulics have been reworked, so that it is decided early to only show menue for chioces, which are really possible.
- LSE and symlink are now linked with ASLR.
- Tranparency glitches in the Junction overlay icon have been fixed.
- Hardlink Clone and Symboliclink Clone have been extended so that Inner Junctions and inner Symbolic Links are properly handled.
- LSE now supports also the replacement of Mountpoints and Symbolic Links.
- LSE shows a dialog box whether explorer should be restarted or not during non silent installation.
- Under Windows Vista and Window7 the *Delete Junction* menue doe not show up anymore.

July 19th 2010 Version 3.3.5.8

- LSE now deals with Symboliclinks during Smart Copy.
- LSE supports Smart Move functionality, which updates inner junctions/symlinks in case of moving/renaming directories
- Added localization for Brazilian Portuguese. Thanks to Marcio R. for the translation.
- Fixed flaws of the automatically renaming feature with respect to directories under W7.
- Overriding custom overlay icons under HKCU was flawed.
- Hardlink Clone now restores the attributes of cloned folders.
- Smart Move progress bar showed a wrong caption text.
- Added Polish localisation, Thanks to Arthur from Poland.
- Fixed a crash during undeleting files from Recycle Bin.
- Mountpoints could not be properly created under Windows XP.
- gFlags was not properly read from the HKCU registry, causing *Smart Move Disable* and *Remote capabilities Disable* to malfunction.
- Fixed the problem of Hardlink Clone stopping unsolictedly after about 500msec.

February 21st 2010 Version 3.2.2.4

- Added localization for Chinese and Russian. Thanks to Zuo Weiming and Ivan(b0s) for the translations.
- Longer lasting operations, like Smart Copy, Symbolic Link Clone, Hardlink Clone, or Enumerate Siblings show a progress bar.
- The Properties dialog of an item offers an 'Explore Target' button for Junctions, Mountpoints and Symbolic Links.
- Added localization for Japanese. Thanks to Taka from Japan!
- Under Windows7 auto rename behaves in the same way as Windows7/8 does for '- Copy'.

September 28th 2009 Version 3.1.6.0

- With W2K the junction creation was broken.
- Fixed a handle leak caused by enumerating hardlink siblings under non Vista/W2K8
- Under Vista & W7 one can create junctions everywhere without elevation, but not in e.g c:\Program Files. LSE is now aware of this and asks elevation for junction creation when neccessary.
- With drives mapped via a Remote Dektop session, the whole explorer & remote desktop session hang, when this drives was expanded in explorer, but only under W2K3 as terminal server.
- Support for Windows7
- Hardlink Sibling Enumeration now also works for XP, W2K NT4, but due to OS contraints not that fast as with Windows7.
- Under W2K it turned out, that CreateHardlink() from kernel32.dll with long pathnames (e.g. \\?\) was broken.
- Fixed a memory leak in serving as COM server.

October 4th 2008 Version 3.0.0.1

- There is a new Smart Copy feature, which enables LSE to copy whole folder structures and preserve the inner hardlink and junction structure.
- Very Long Pathname support added for Smart Copy and Hardlink Clone.
- Junctions can now be created targeting also Junctions.

June 21st 2008 Version 2.9.5.3

- Hardlinks can be enumerated under Vista & Windows7.
- Fixed a handle leak for HKCU\Software\LinkShellExtension.
- Removeable media support didn't work, when remote capabilities were switched off.

May 1st 2008

Version 2.9.0.3

- Already existing Junctions can be replaced by dragging a directory over it
- Naming has been streamlined more towards 'Link Shell Extension'
- If the maximum number 1023 of hardlinks for a file is exceeded, an error message is displayed. This applies for hardlinks and hardlink clones.
- The Vista & Windows7 junction overlay icon in 256x256 is in proper size.
- Custom icons can be specified for Junction and Hardlink overlays
- Version for Itanium available
- Pick/Drop does not interfere with the creation of Hardlinks, Junctions or Symbolic Links via Drag and Drop. It is now possible to pick a link, then drag another file via right mouse click to some location, drop it there and

afterwards drop the first, picked file
- The property dialog of a Volume Mountpoint now displays the logical drive letter of the mounted drive instead of the odd volume name.
- Ongoing work towards localisation to East Asian languages.
- LSE now also works on removable NTFS media, which are not A: or B:
- The location in the registry to specify the LSE language has changed, since the old place under HKCR was not Vista & Windows7 compatible at all.
- Default values for language settings and overlay icons settings are copied over automatically to freshly logged on users profile
- Volume Mountpoint support for Vista & Windows7.
- Introduced silent install capabilities.
- Symbolic Links now can be created even if the filename contains UTF-16(Asian)characters.
- LSE now also works for non Administrators under Vista & Windows7 (after they acknowledged the elevation dialog with the admin password for sure).
- Symbolic Links for files or directories can now be created across volumes.
- LSE now can also create Hardlinks from Shortcuts, which didn't work for ages.
- The print name (the name some can see to the right of a junction, after issuing 'dir' in a command prompt) for Junctions under Vista & Windows7 is now correct.
- Lots of usability fixes
- During Installation under Vista64 Explorer automatically gets restarted.
- The setup contains a check if the VS2005 SP1 Redistributable Package is installed.
- The setup contains a check if the LSE version for the proper platform is going to be installed.
- Vista & Windows7 compliant overlay icons for Hardlinks.

January 20th 2008 Version 2.8.0.6

- Hardlinks show up with a small overlay icon. This icon will change for Vista compliancy, but at least it is here now.
- Junctions have a Vista compliant overlay icon in many resolutions
- Support of creation and deletion of Volume Mountpoints. Unfortunately this does not work under Vista
- LSE now prevents the creation of 'loops', when setting up a junction or hardlink clone
- Hardlink 'cross drive drops' are now not possible anymore
- Some bug fixes for NT4
- Lots of usability fixes

October 16th 2007 Version 2.7.1.0

- First fixes for the x64 world. Maybe more to come. Fixing x64 is top priority since I have Q6600 myself now...

March 25th 2007 Version 2.7.0.1

- Fixed a nasty bug, which caused HardlinkShellExt to slow down explorer, when it was started. Also fixed the problem that it accessed drive A:, when an explorer started.
- PropertySheet on file and directory properties will show various info with W2K/XP
- *Delete Junction* is back, because sometimes, especially when deleting junctions, which point to directories with big amount of data, the Copyhook Handler does not act as expected. Until this phenomenon is solved, *Delete Junction* is back.

January 12th 2007 Version 2.6.0

- Link Shell Extension is now robust with respect to deleting junctions. Delete commands issued from explorer unlink junctions, but do not delete its content.
- Due to Junction-awareness of explorer the *Delete Junction* is gone from the context menu
- Support for Windows Vista & Windows7. Link Shell Extension is now capable of creating Symbolic Links, but also has a few restrictions related to the Reference column

December 27th 2006 Version 2.5.1 released

- Added localisation for Italian and Spanish to commands and messages. Thanks to Nicola Guidotto and Diego Segobia for the translations.

December 6th 2006 Version 2.4.0 released

- Added localisation for French and German to commands and messages

November 26th 2006 Version 2.3.0 released

- Minor fixes in the installer/deinstaller
- Introduced the hydraulics of multiple auto rename. If you drop links/junctions in the same directory, now it behaves exactly as explorer and puts numbers on the multiple instantiations of a file.

June 16th 2006 Version 2.2.2 released

- BugFix. Link Shell Extension is now also able to create hardlinks via Drag and Drop in root dirs of a drive.

May 29th 2006 Version 2.2.1 released

- Updated documentation after review of Philip Daniels
- Made a big step forward to being Vista compliant
- Fixed path length limitations in several places
- Junctions can span over local NTFS volumes

March 14th 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 context menu, when more than one entries would be added to context menu to show the many dropping choices
- Support for SymbolicLinks with 'Vista'.
- Support for SymbolicLink 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 cancelled now.
- The installer restarts explorer.exe to properly add/remove the shell extension
- Added an entry to Start Menu/Programs
- Support for WindowsXP64.

November 26th 2005 Version 1.7 released

- Added the *Delete Junction* context 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 doc 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 occurrence 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 read-only 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 3.7.5.x version is a stable version for the supported plattforms. Especially for Vista & Windows7 the reference count column stuff will be addressed in one of the upcoming releases.

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.
Nullsoft for the great lean and mean nsis installer
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 tips on junction
Philip Daniels for a technical writers documentation review
Daniel Thibault for the French localisation, and for a dozen of bug reports and feature requests.
Masatoshi Kimura for the symbolic link driver for WindowsXP. localisation, and for a dozen of bug reports and feature requests.

Open Issues

- With Vista & Windows7 the column handler in explorer, providing the reference count, does not work, since Microsoft deprecated the interfaces with respect to this functionality.

License

- This program is provided as is. See license.txt from this distribution for legal issues.
- Link Shellextension uses tre as the regular expression machine. See the tre license.
- ln.exe uses the Rockall v4.0 heapmanager for fast heap operations. See the EULA.

Contact / Donations

Bug reports, or feature requests send to Hermann Schinagl..

LSE is and will be freeware, but if LSE was really helpful for you and saved lots of your time please think of donations either via PayPal

Donate

or by flattring me

or by sending me a gift certificate from

or by donating bitcoins:

19irqhz5cMDkp2hf9YWqDY26PgYguJQdc7

Link Shellextension also has its page on Facebook, where you can find announcements for new releases, and you can discuss feature requests

Link Shellextension broadcasts its release notes via RSS.

Download

Windows XP64
Windows Vista64
Windows 7 64bit
Windows 8 64bit

This version contains the 64bit version of Link Shell Extension, but also contains a 32bit version, which is installed in paralell to the 64bit version, to satisfy third party filemanagers/explorers like total commander.
Please make sure that the necessary runtime .dlls for 64bit and 32bit are installed on your system.

The x64 prerequisites package can be downloaded from Microsoft:

vcredist_x64.exe for VS2005 SP1, version 6195/June 2011 (3.0 Mb)

The x86 prerequisites package can be downloaded from Microsoft:

vcredist_x86.exe for VS2005 SP1, version 6195/June 2011 (2.6 Mb)

Afterwards install the

Link Shell Extension (3.68Mb)

Windows 2000
Windows XP
Windows Server 2003
Windows Server 2008
Windows Vista
Windows 7
Windows 8

Please make sure that the necessary runtime .dlls are installed on your system. This prerequisites package can be downloaded from Microsoft:

vcredist_x86.exe for VS2005 SP1, version 6195/June 2011 (2.6 Mb)

Afterwards install the

Link Shell Extension (3.44Mb)

Windows Itanium
Windows 7 Itanium
Windows 8 Itanium

Please make sure that the necessary runtime .dlls are installed on your system. This prerequisites package can be downloaded from Microsoft:

vcredist_IA64.exe for VS2005 SP1, version 6195/June 2011 (6.3 Mb)

Afterwards install the

Link Shell Extension (3.59Mb)

Windows NT4

The version for Windows NT will be no more activley developed on, and its functionality is frozen with version 3.2.0.0, which basically has all the important features.

Link Shell Extension (1.13Mb)

Symbolic Link Driver for Windows XP

The driver to enable even WindowsXP with symbolic link functionality is provided courtesy of Masatoshi Kimura. You can download the driver from his homepage or from my site acting as a mirror.

Symbolic Drivers for WindowsXP 64 (86kb) [Original Location]
Symbolic Drivers for WindowsXP 64 (86kb) [Mirror schinagl.priv.at]

Symbolic Drivers for WindowsXP (86kb) [Original Location]
Symbolic Drivers for WindowsXP (86kb) [Mirror schinagl.priv.at]

Sources for Drivers (23kb) [Original Location]
Sources for Drivers (23) [Mirror schinagl.priv.at]