# WIN_LNK 1.18 27.Nov.20

## Inhaltsverzeichnis

# 1 Overview

Win_Lnk served originally only to use the Windows links under MagicPC as real symbolic links. More and more other MagicPC extensions are also inserted. To do that some GEM-DOS-and AES routines are adjusted. In order to bend the BS routines Trapper is used.

## 1.1 Foreword to version 1.18

After more than 10 years, a user contacted me about problems with the clipboard. These were known to me myself, but since I rarely use MagicPC, I ignored them. But now there was a reason to do it. The first thing I noticed was that I was using version 1.17, but had never published it and not documented it. So I have now solved the problem with the clipboard and tried to understand the changes from 1.17 and to improve them if necessary

# 2 State of development

This is one of the first public versions, so there still might be some errors.

# 3 Installation

For installation, the unlnk.dll must be copied into the MPS directory (create if necessary) in the MagicPC folder. The Win_Lnk.prg actually belongs in the Auto folder, Start folder, Autoexec.bat or you can register it as Autostart via "Register application". The latter has the advantage that it can use AES functions, e.g. shel_envrn (to query the environment variable for the Windows extensions) or scrp_read (to determine the path of the clipboard). The start folder could work too. But it can also be started normally. In any case, Trapper must be started beforehand!
Win_lnk also includes a win_lnk.ini.

## 3.1 win_lnk.ini

This file normally belongs in the same folder as win_lnk.prg, or when started from autoexec.bat or similar. into the root directory of the boot partition, usually to C: \. Here you can deactivate the individual extensions if necessary. Unless otherwise specified, win_lnk only evaluates the 1st and 3rd characters of a line. The 1st determines the function and the 3rd determines the state. 0 is used to
deactivate and anything else to activate it. Everything that is not explicitly switched off is switched on.
No rule without exception, as the transfer of scrap.rtf does not work perfectly in all situations, this is switched off by default and must therefore be activated explicitly with a 1
A line beginning with % is a comment. Some examples
a:0 switches off the (non-existent) function a
a:1 or
adhgdlsk turns them on. But it's better to limit yourself to a:0 and a:1
But
r:2 deactivates the transfer of scrap, rtf
r:1 activates it
w is a special case, s. Starting Windows programs
In the section The extensions in detail, the individual points are each given the identifier in brackets.

# 4 Deinstallation

If you restart Win_Lnk when it is already active,you can deinstall it. But this might take time, if some reserved memory is still in use (Pexec,...). This is only true, if both instances of win_lnk have the same version number. But if for example a win_lnk 1.05 is running and you start a 1.06, than 1.05 is replaced by 1.06

# 5 The extensions  in detail

## 5.1 Windows-Links (L)

Use Windows links under MagicPC as full symbolic links. The purpose was that MagicPC behaves as much as possible as Magic. This, however, in such a way that it is based on the possibilities of Windows. How to use symbolic links under MagicPC is explained in the Magic documentation. However, there are some specific differences. You can define links under Windows or MagicPC Some things wich are possible under Windows would not be possible under Magic and thereby might couse problems. The main problem are doubly defined names. For further details see. under Dreaddir(297) und Dxreaddir (322). My advice, omit such things.

## 5.2 Environment variables (E)

Win_lnk offers a possibility to define Environment variables under Windows and to use them under MagicPC. This is mainly usefull with multiuser-Windows variants, because one can define the variable there differently for every user. Details see under Pexec(75) und shel_envrn(AES 125)

## 5.3 Starting Windows programs (x und w)

To start Windows programs from MagicPC there is MPC_ACC.ACC. However, this has the disadvantage that it forgets to adjust the paths. Windows program gets the MagicPC - paths. Now WIN_LNK offers the possibility to run Windows programs like normal programs. The ACC is required no more. By defining a string in win_lnk.ini you can determine which programs WIN_LNK should ask Windows to execute. Up to version 1.16 this was done using an environment variable. When started from the Auto folder, however, this cannot be read. For reasons of compatibility, however, it will still be tried. The INI file has priority. If only * .exe is to be transferred to Windows, then put:
w:.EXE;
in the ine file. To add more extensions these are simply added seperated by a point. In addition, you have to tell your Shell that it should treat these files like programs. This happens in Jinnee, e.g., under Other / settings. There select on the left side programs and add the extension, e.g., *.exe. as GEM-Programs (I never saw Jinnee in english)

## 5.4 clipboards (s)

MagicPC can connect the clipboards on Atari side with the Windows side. Unfortunately, this functions in the direction Atari-> Windows only once. This can be correct by WIN_LNK. Up to now WIN_LNK only supports( like MagicPC )  texts. Win_lnk ignores the setting of MagicPC if the Windows - clipboard should be used. Who does not want it it must deactivate it at both places.

### 5.4.a clipboards RTF ( r )

Some programs also write scrap.rtf, i.e. formatted text. This can also be transferred to the Windows clibboard as a auch. In order for it to work as it should, some conditions must be met. So the word processor or similar must. first write scrap.txt and then scrap.rtf, only then will both end up in the clipboard, but this works with the programs tested so far. Anything else would be very complicated to implement. This is just an addition to the previous one, so if s:0 is set, r:1 is ignored As already written under installation, only a 1 in the third position is recognized in the ini file for activation. So e.g. r:1

## 5.5 Égale (k)

Égale: So that Égale has fewer problems with the case sensitivity of filenames, I patched it, but also changed OS functions, these can either be installed via the separate dir_lwr ior here. Further information at  Dir_lwr

# 6 For programmers

## 6.1 Logfile

From V 0.32 a log file can be written from the Windows side. This Function can be (de)activated in two ways:

1. by defining DOLOG (#define DOLOG) both in win_lnk.c and in unlnk.cpp.
2. by restarting Win_Lnk, there you can also deinstall Win_Lnk.

From version 1.0 to 1.16, DOLOG was no longer defined by default, but anyone who wants can easily reactivate it in the source code. Simply define DOLOG. The reason it got blown was because this feature caused as many errors as it helped solve. This has been solved since 1.17 → DOLOG is defined again For the loging to work either the system variable LOGDIR or TEMP must be defined (on the Windows side!)! Alternatively one can also debug MagicPC ( Task manager.) Then, if DOLOG is definded but the loging is disabled, TRACE prints into the debug window. Because some MagicPC versions have problems with this Debug-dll, I add only the release version. To the reactivate the log one must recompile the dll anyhow.

## 6.2 changed functions

In parentheses the Gemdos-function numbers are given. The functions are listed In groups. Single functions can appear several times.

### 6.2.a Fcreate(60) Fopen(61), Pexec(75) and Fattrib(67)

If these functions are applied to a link, this is followed. Indeed only if it has´nt the extension .lnk. Because the Fileselector, Fsfirst/Fsnext, and D(x)readdir the LNK cut that extension off, this is the normal case. In addition, in Pexec Windows environment variables are added, s. afterwords and it can instruct Windows to start Windows programs. Fcreate and Fopen are also changed for transferring of the clipboard see below.

### 6.2.b Pexec(75) and shel_envrn(AES 125)

A problem of MagicPC is that it has no multi user abilities. An important point, are user dependent environments. Above all e.g., HOME. However, depending of the Windows version under which MagicPC runs this might be offered by the host system. So why not just transmit the environments from windows to MagicPC. I´ll explain it most simply by an example. If we would like to define HOME on the Windows side, then add the following line to Magx.inf:

```
#_ENV HOME=%WIN%
```

If HOME is used now, Win_Lnk recognizes that it should take the definition of HOME from Windows. However, if for some reasen HOME should have different values under both systems, then one can choose on Windows any other name, e.g., HOME_MP. Then add the following line to Magx.inf:C.

```
#_ENV HOME=%WIN%:HOME_MPC
```

In the 1st case one must define HOME and in the 2nd HOME_MPC under Windows. Environments can be retrieved under MagicPC in 2 ways, either by shel_envrn that of the Shell, or from the own basepage. The latter is given down with pexec or is defined newly. Therefore, both ways are suported. The syntax must be kept exactly, thus no additional blanks or similar. Win_lnk tries to interprete the variable as a filepath and if necessary transfer it from the Windows path system to the MagicPC System. So define Pathes under Windows as Windows knows them.

### 6.2.c Fcreate(60) Fopen(61) Fclose(62) and Fdelete(65)

if scrap.txt (in the suitable Directory) is opened in write mode(Fopen or Fcreate) win_lnk memorises the handle. If this file is closed later, win_lnk will copy the content to the Windows clipboard. The same applies to scrap.rtf. If this scrap.txt is deleted, the Windows clipboard is deleted accordingly

### 6.2.d Fcntl(260) cmd=0x4600

Here the S_IFLNK bit is set if necessary.

### 6.2.e Fxattr(300)

According to the Flag links are followed or not. Folder links are always followed.

### 6.2.f Freadlink(303)

Returns the target of a link

### 6.2.g Dreaddir(297) and Dxreaddir (322)

If a file is recognized as a link, the extension .lnk will be removed. If then the filename is not unique any more a ~L will be added to the file name. In addition, in mode S_IFLNK is set. Let´s say there are the following 3 files in a directory:

1. file1.txt.lnk
2. file1.txt
3. file2.txt.lnk

if in 1) only the .lnk would be removed, 1) and 2) would not be distinguishable any more.
Therefore, D(x)readdir changes the names to:

file1~L.txt
file1.txt
file2.txt

### 6.2.h Dopendir(296) and Dclosedir(299)

These both functions react as before, but since D(x)readdir needs some parameter from Dopindir, I save them here. And since this memory has to be freed sometime, I also had to change Dclosedir.

### 6.2.i Fsnext(79) and Fsfirst(78)

The goal was to return short file names (8.3) for links which however have the right extension. The normal Fsnext would return always a *.lnk. A problem are files which exist once as normal files and once as a link. In addition, the change of the extension had to be reversible. Here an example:
In a directory there are the following files:

1. liesmich.txt
2. liesmich.txt.lnk
3. liesmich
4. liesmich.lnk
5. readme
6. readme.lnk
7. lizemoi.txt.lnk
8. lizemoi.lnk
9. cat.prg.lnk

All lnk files are links. The normal Fsfirst would return these 9 names:

1. liesmich.txt
2. liesmi~1.lnk
3. liesmich
4. liesmich.lnk
5. readme
6. readme.lnk
7. lizemo~1.lnk
8. lizemoi.lnk
9. catprg~1.lnk

The file 2) would´nt be recognizable for an editor as a text file, and the files 4), 6), 7) 8) and 9) would have extensions meaningless to most users and programs. Therefore, the new Fsfirst/Fsnext routine return these names:

1. liesmich.txt
2. liesmi~B.txt
3. liesmich
4. liesmich.~L
5. readme
6. readme~L
7. lizemo~B.txt
8. lizemoi.~L
9. catprg~B.prg

Should these files be opened later, e.g., with Fopen, Win_Lnk recognizes the ~ followed by a letter and knows that these are accordingly manipulated file names. And it can reconstruct the original 8.3 short names. "~L" and ". ~L " will get simply be eliminated ~A to ~J will be replaced by ~0 to ~9. In addition, if an extension exists it will be replaced by lnk or otherwise .lnk will be added. What I realy don´t like is case 4) but methode 6) is not aplicable for files with nomes of more than 6 chars. 9) is also not very nice.

### 6.2.j fsel_input (AES 90) and fsel_exinput(AES 91), rsrc_load (AES 110), shel_find(AES 124) and shel_write (AES 121)

have automatically adapted themselves so to speak, they probably internaly call other changed Gemdos functions.

### 6.2.k Fdelete(65), Frename(86)

If source file is a link, this one (the lnk file) will be deleted or renamed.

### 6.2.l Fsymlink(302)

Here I have wrote a all new routine which accesses the Windows routines. The link target is followed if necessary. So no links on links are created. This is the way Windows does it, and because MagicPC should follow the rules of Windows in it´s file system, I just did it in the same way.

### 6.2.m Dpathconf(292)

here only directory links are followed.

### 6.2.n Dcreate(57), Ddelete(58) und Dsetpath(59)

If these return an EPTHNF error it will be tried to follow a link and then if necessary to call the function again.

## 7 Testimonials

I still use MagicPC and thus win_lnk, but rarely. So don't get annoyed about errors but tell me about them. The probability that I will notice them myself is slim. If something doesn't work, the log file will help me.If the system variable LOGDIR is defined under Windows, it will be stored there, otherwise under TEMP (usually C: \ Users \ User \ AppData \ Local \ Temp where Windows probably always translates the first user) The file itself is called win_lnk.log

# 8 Changes since version 1.17

1.17 around 2006 but never published

- There were problems with WinExe because win_lnk usually cannot use AES, so the list of extensions must now be defined via the ini, not as Environment s. 3)  but see V1.18 below
- Fixed problems with the links
- Memory management revised
- The changes in Pexec only make sense in some modes, but have been tried in all.
- The information whether Dreaddir has to convert to lowercase letters is packed into the DIR structure on the WIN side (for the Égale extension)
- If necessary, error message at the end of the program
- New log file output
- internal changes

1.18 from 27.Nov.2020

- The transfer of the clipboard to Windows did not work if no scrp_write was executed.
- New recommendation not to start win_lnk via the auto folder (autoexec.bat) but to register the application
- If you start win_lnk manually or as a registered application, you can also use the definition of the Windows extensions via environment variable.
- win_lnk can now also pass formatted text (rtf) to Windows. The other direction does not work because this would have to be done by MagicPC. This is in the ß-stage

Dimitri Junker