WIN_LNK 1.18 27.11.20

Inhaltsverzeichnis

1 Überblick

Win_Lnk diente ursprünglich nur dazu die Windows-Verweise unter MagicPC als vollwertige symbolische Links zu verwenden. Mit und mit werden aber auch andere MagicPC-Erweiterungen eingebaut.Dazu werden einige GEM-DOS- und AES-Routinen angepaßt. Zum Verbiegen der BS-Routinen wird Trapper benutzt.

1.1 Vorwort zur Version 1.18

Nach über 10 Jahren meldete sich ein Nutzer wegen Problemen mit der Zwischenablage. Diese waren mir selber bekannt, da ich aber MagicPC immer wenige nutze hatte ich es ignoriert. Aber jetzt gab es einen Grund es anzugehen. Dabei fiel mir als erstes auf, daß ich die Version 1.17 nutte, aber nie veröffentlicht hatte und auch nicht dokumentiert. Ich habe jetzt also das Problem mit der Zwischenablage gelöst und versucht die Änderungen aus 1.17 nachzuvollziehen und ggf. Noch nachzubessern

2 Entwicklungsstand

Dies ist eine der ersten öffentlichen, also nicht ß-Versionen, aber garantieren, daß keine Fehler auftreten kann ich leider nicht.

3 Installation

Zur Installation muß die unlnk.dll in das MPS Verzeichnis (ggf. anlegen) im MagicPC-Ordner kopiert werden. Das Win_Lnk.prg gehört eigentlich in den Auto-Ordner, Start-Ordner, Autoexec.bat oder man meldet es als Autostart per „Anwendung anmelden“ an. Letzteres hat den Vorteil, daß es AES-Funktionen nutzen kann, also z.B. shel_envrn (um die Environmentvariable für die Windowsendungen abzufragen) oder scrp_read (um den Pfad der Zwischenablage zu ermitteln) nutzen kann. Der Start-Ordner könnte auch funktionieren. Es kann aber auch normal gestartet werden. Auf alle Fälle muß Trapper vorher gestartet werden!
Zu win_lnk gehört auch ein win_lnk.ini.

3.1 win_lnk.ini

Dieses File gehört normalerweise in den gleichen Ordner wie win_lnk.prg, bzw bei Start aus autoexec.bat o.ä. ins Rootverzeichnis der Bootpartition, also normalerweise nach C:\. Hier kann man die eninzelnen Erweiterungen bei Bedarf deaktivieren. Soweit nicht anders angegeben wertet win_lnk nur das 1. und 3. Zeichen einer Zeile aus. Dabei bestimmt das 1. die Funktion und das 3. den Zustand. Dabei wird 0 als aus und alles andere als ein akzeptiert. Alles nicht explizit abgeschaltete wird eingeschaltet.
Keine Regel ohne Ausnahme, da die Übertragung von scrap.rtf nicht in allen Situationen perfekt funktioniert ist dies defaultmäßig abgeschaltet und muß also explizit mit einer 1 aktiviert werden
Eine Zeile die mit % beginnt ist Kommentar. Einige Bsps
a:0 schaltet die (nicht vorhandene) Funktion a aus
a:1 oder
adhgdlsk schaltet sie ein. Aber besser man beschränkt sich auf a:0 und a:1
Aber
r:2 deaktiviert die Übertragung von scrap,rtf
r:1 aktiviert sie
w ist ein Sonderfall, s. Starten von Windows Programmen
Im Abschnitt Die Erweiterungen im einzelnen stehen bei den Einzelnen Punkten jeweils die Kennung in Klammern.

4 Deinstallation

Startet man Win_Lnk obwohl es bereits installiert ist, so kann man es deinstallieren. Dies geht aber ggf. nur Schrittweise, da evtl. noch Speicher freigegeben werden muß. Dies kann beliebig lange dauern. Wenn z.B. für Pexec Speicher reserviert wurde kann der erst freigegeben werden wenn das Programm beendet wird.Dies gilt aber nur wenn beide Instanzen die gleiche Versionsnummer haben. Läuft aber z.B. win_lnk 1.05 und man startet 1.06 so beendet sich 1.05 und die 1.06 wird installiert.

5 Die Erweiterungen im einzelnen

5.1 Windows-Verweise (L)

Windows-Verweise unter MagicPC als vollwertige symbolische Links verwenden. Das Ziel war, daß sich MagicPC bei Links möglichst genau so verhält wie Magic. Dies aber so, daß es tranparent auf den Möglichkeiten von Windows aufbaut.Wie man unter MagicPC symbolische Links verwendet steht in der Magic Anleitung. Es gibt aber einige Besonderheiten. Links anlegen kann man unter MagicPC oder unter Windows. Dabei sind unter Windows Sachen möglich die unter Magic nicht möglich wären und dadurch zu Problemen führen. I.W. geht es dabei um doppelte Namen. Genaueres s.u. unter Dreaddir(297) und Dxreaddir (322). Mein Rat, so etwas einfach unterlassen.

5.2 Environmentvariablen (E)

Win_lnk bietet eine Möglichkeit an Environmentvariablen unter Windows zu definieren und unter MagicPC zu nutzen. Dies ist vor allem sinnvoll bei Multiuser-Windowsvarianten, da man dort für jeden User die Variable anders definieren kann. Details s.u. unter Pexec(75) und shel_envrn(AES 125)

5.3 Starten von Windows Programmen (x und w)

Zum Starten von Windows Programmen aus MagicPC heraus gibt es MPC_ACC.ACC. Dies hat aber den Nachteil, daß es vergißt die Pfade anzupassen. Das Windows Programm bekommt also die MagicPC - Pfade. WIN_LNK bietet nun die Möglichkeit Windowsprogramme wie normale Programme zu starten. Das ACC wird also nicht mehr benötigt. Durch definieren eines Strings in der win_lnk.ini kann man bestimmen welche Programme WIN_LNK durch Windows ausführen lassen soll. Bis Version 1.16 geschah dies durch eine Environment-Variable. Bei Start aus dem Auto-Ordner kann diese aber nicht gelesen werden. Aus Kompatibilitätsgründen wird es aber weiterhin versucht. Vorrang hat aber die INI. Soll also nur *.exe an Windows übergeben werden, so muß dort:
w:.EXE;
stehen. Um weitere Endungen hinzuzufügen werden diese einfach jeweils durch einen Punkt getrennt angefügt. Außerdem muß der verwendeten Shell klar gemacht werden, daß sie diese Files wie Programme behandeln soll. Dies geschieht in Jinnee z.B. unter Sonstiges/Einstellungen. Dort links Programme auswählen und unter GEM-Programme die Endung, also z.B. *.exe, hinzufügen.

5.4 Zwischenablage (s)

Eigentlich kann MagicPC die Zwischenablagen auf Atari Seite mit der Windowsablage verbinden. leider funktioniert dies in der Richtung Atari -> Windows nur einmal. Dies kann WIN_LNK korrigieren. Bisher kümmert sich WIN_LNK (wie MagicPC überhaupt) nur um Texte. Win_lnk ignoriert die Einstellung von MagicPC ob die Windows - Zwischenablage benutzt werden soll. Wer es also nicht will muß es an beiden Stellen abstellen.

5.4.a Zwischenablage RTF (r)

Einige Programme schreiben auch ein scrap.rtf, also formatierten Text. Dieser kann hiermit auch als soclher an die Windowsablage übertragen werden. Damit die funktionieert wie es soll müssen einige Bedingungen erfüllt sein. So muß die Textverarbeitung o.ä. zuerst scrap.txt und dann scrap.rtf schreiben, nur dann landen beide in der Zwischenablage, dies funktioniert aber mit den bisher getesteten Programmen. Alles andere wäre sehr kompliziert zu implementieren. Dies ist nur ein Zusatz zum vorigen, ist also s:0 gesetzt wird r:1 ignoriert Wie schon unter Installation geschrieben wird im ini File nur eine 1 an dritter Stelle zum aktivieren erkannt. Also z.b. r:1

5.5 Égale (k)

Égale: damit Égale weniger Probleme mit der Groß/Kleinschreibung von Filenamen hat habe ich es gepatcht, dazu gehörn aber auch noch geänderte BS-Funktionen, diese kann man entweder über das separate dir_lwr instalieren oder eben hier. Weitere Infos bei Dir_lwr

6 Für Programmierer

6.1 Logfile

Ab V 0.32 kann von der Windowsseite ein Logfile geschrieben werden. Dise Funktion ist in 2 Stufen (de)aktivierbar:

1. durch definieren der Konstanten DOLOG (per #define) (Sowohl in Win_Lnk.c als auch im unlnk.cpp)
2. durch erneuten Start von WIN_LNK. Hier hat man dann auch die Möglichkeit Win_lnk ganz zu beenden.

Von Version 1.0 bis 1.16 war DOLOG standarmäßig nicht mehr definiert.Wer will kann es aber im Quelltext leicht reaktivieren. Einfach DOLOG definieren . Der Grund, daß es rausflog war, daß diese Funktion ebensoviele Fehler verursachte wie sie half zu lösen. Dies ist seit 1.17 gelöst → DOLOG ist wieder gesetzt
Damit das mitlogen funktioniert muß die Systemvariable LOGDIR oder TEMP definiert sein (auf Windowsseite!)!
Alternativ kann man auch MagicPC debuggen (im Taskmanager rechts anklicken...) Dann werden bei abgeschaltetem Logfile (aber definiertem DOLOG) TRACE-Ausgaben in das Debugfenster gemacht. Da einige MagicPC Versionen mit dieser Debug-dll aber Probleme haben lege ich nur die Release-Version bei. Zur Reaktivierung der Logausgabe muß man die dll ja sowieso neu erstellen..

6.2 Veränderte Funktionen

In Klammern sind jeweils die Gemdos-Funktionsnummern angegeben Die Funktionen sind Gruppenweise aufgeführt. Einzelne Funktionen können mehrfach auftreten.

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

Werden diese Funktionen auf einen Link angewendet, so wird dieser verfolgt. Allerdings nur wenn die Endung .lnk nicht angegeben wird. Da der Fileselector, Fsfirst/Fsnext, und D(x)readdir das LNK abschneiden ist dies der Normalfall. Bei Pexec werden außerdem Windowsenvironments ermittelt, s. hiernach und es kann das Starten von Windowsprogrammen an Windows übergeben. Fcreate und Fopen werden auch zum Übertragen der Zwischenablage verändert s.u.

6.2.b Pexec(75) und shel_envrn(AES 125)

Ein Problem von MagicPC ist, daß es keine Multiuserfähigkeiten hat. Ein wichtiger Punkt dabei sind userabhängige Environments. Vor allem z.B. HOME. Je nach Windowsversion unter MagicPC bietet dies aber das Hostsystem. Was liegt also näher als Environments durchzureichen. Ich erkläre es am einfachsten an einem Bsp. Nehmen wir an, ich möchte HOME auf Windowsseite definieren, dann setze ich in Magx.inf die folgende Zeile:

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

Wird jetzt HOME abgefragt, so erkennt Win_Lnk, daß es als Resultat die Definition von HOME auf Windowsseite nehmen soll. Angenmmen es gäbe aber auf Windowsseite auch ein HOME und beide sollen verschieden sein, dann kann man auf Windowsseite einen beliebigen anderen Namen wählen, z.B. HOME_MPC. Dazu setzt man in Magx.inf die folgende Zeile:

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

Im 1. Fall muß man also unter Windows HOME im 2. HOME_MPC definieren. Environments können unter MagicPC auf 2 Arten abgefragt werden, entweder per shel_envrn das der Shell, oder über die eigene Basepage. Letzteres wird bei pexec vererbt bzw neu definiert. Deshalb werden beide Befehle angepaßt. Die Syntax muß genau eingehalten werden, also keine zusätzlichen Leerzeichen o.ä. Win_Lnk versucht den von Windows erhaltenen Parameter als Pfad zu interpreteren und in das MagicPC Filesystem zu übertragen, schlägt dies fehl wird der String unverändert übergeben. Also Pfade so definieren wie sie unter Windows richtig sind.

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

wird scrap.txt (im entsprechenden Direktory) zum schreiben geöffnet (Fopen oder Fcreate) so merkt sich win_lnk das Handle. Wird dieses File später wieder geschlossen, so kopiert win_lnk den Inhalt in die Windowsablage. Entsprechendes gilt für scrap.rtf. Wird dieses scrap.txt gelöscht wird entsprechend die Windows-Zwischenablage gelöscht

6.2.d Fcntl(260) cmd=0x4600

Hier wird ggf. das S_IFLNK Bit gesetzt

6.2.e Fxattr(300)

Je nach Flag werden Links verfolgt oder nicht. Ordnerlinks werden immer verfolgt.

6.2.f Freadlink(303)

Gibt das Ziel eines Links zurück

6.2.g Dreaddir(297) und Dxreaddir (322)

Wird ein File als Link erkannt, wird das .lnk entfernt Ist der Filename dann nicht mehr eindeutig ein ~L an den Filenamen angehängt. Außerdem wird in mode S_IFLNK gesetzt. Bsp es gebe in einem Dir folgende 3 Files:

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

würde bei 1) einfach das .lnk entfernt, wäre 1) und 2) nicht mehr unterscheidbar.
Deshalb macht D(x)readdir daraus:

```
file1~L.txt
file1.txt
file2.txt
```

6.2.h Dopendir(296) und Dclosedir(299)

Diese beiden Funktionen reagieren wie gehabt, allerdings benötigen D(x)readdir Parameter von Dopendir, deshalb speicher ich diese hier zwischen. Und da dieser Speicher auch irgendwann wieder freigegeben werden soll mußte auch Dclosedir verändert werden.

6.2.i Fsnext(79) und Fsfirst(78)

Das Ziel ist, kurze Dateinamen zurück zu geben, die bei Links aber die richtige Endung haben. Das normale Fsnext würde immer *.lnk zurückgeben Probleme machen Files die einmal normal und einmal als LNK existieren. Außerdem muß die Änderung der Endung später wieder rückgängig gemacht werden können. Zur Erklärung ein Bsp:
In einem Dir seien die folgenden Dateien:

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

Alle lnk Files seien Links. Das normale Fsfirst würde aus diesen 9 Namen so etwas machen:

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

Die Datei 2) wäre also für einen Editor nicht mehr als Textfile erkennbar, und die Dateien 4),6),7) 8) und 9) hätten Endungen mit denen die wenigsten Programme und User etwas anfangen könnten. Deshalb machen die neuen Fsfirst/Fsnext Routine daraus:

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

Sollen diese Dateien später z.B. mit Fopen geöffnet werden, so erkennt Win_Lnk an den ~ gefolgt von einem Buchstaben, daß es entsprechend manipulierte Dateinamen sind. Und es kann die ursprünglichen auf 8.3 gekürzten Namen rekonstruieren. "~L" und ".~L" werden einfach weggelassen ~A bis ~J werden durch ~0 bis ~9 ersetzt. Außerdem wird eine vorhandene Endung durch lnk ersetzt bzw .lnk angehängt. Am wenigsten gefällt mir 4), aber bei Filenamen >6 ist die Methode von 6) leider in den Grenzen von 8.3 nicht mehr durchführbar. Auch 9) ist etwas gewöhnungsbedürftig.

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

haben sich sozusagen automatisch angepaßt, sie rufen wohl intern geänderte Gemdos-Funktionen auf.

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

Handelt es sich bei der Quelle um einen Link, so wird dieser (also das lnk-File) gelöscht bzw umbenannt.

6.2.l Fsymlink(302)

Hier habe ich eine eigene Routine geschrieben, die auf die entsprechenden Windows Routinen zugreift. Das Linkziel wird ggf. verfolgt. So entstehen keine Links auf Links. Dies macht Windows genau so, und da MagicPC sich ja im Windowsfilesystem an dessen Regeln halten sollte mach ich das eben auch so.

6.2.m Dpathconf(292)

hier werden nur Ordnerlinks verfolgt.

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

Falls diese ein EPTHNF zurück geben wird versucht den Pfad zu dereferenzieren und dann ggf die Funktion nochmal aufgerufen.

7 Erfahrungsberichte

Ich selber nutzte MagicPC und somit win_lnk zwar noch, aber eher selten Also nicht über Fehler ärgern sondern sie mir mitteilen. Die Wahrscheinlicjhkeit, daß ich sie selber bemerke ist gering. Falls irgendwas nicht funktionieert wird mir das log File helfen, Falls unter Windows die Systemvariable LOGDIR definiert ist wird es dort abgelegt sonst unter TEMP (meist C:\Users\User\AppData\Local\Temp wobei Windows wohl immer das erste User übersetzt) Das File selber heßt win_lnk.log

8 Änderungen seit Version 1.17

1.17 etwa 2006 aber nie veröffentlicht

- bei WinExe gab es Probleme da win_lnk meist kein AES verwenden kann, deshalb muß jetzt die Liste der Endungen über das ini definiert werden, nicht als Environment s. 3) s.a. aber V1.18
- Probleme mit den Links behoben
- Speicherverwaltung überarbeitet
- Die Änderungen bei Pexec sind nur bei einigen Modi sinnvoll, wurden aber bei allen versucht.
- Die Info ob Dreaddir in Kleinbuchstaben wandeln muß mit in die DIR-Struktur auf WIN-Seite gepackt (für die Égale Erweiterung)
- Ggf Fehlermeldung bei Programmende
- Neue Logfileausgabe
- interne Änderungen

1.18 vom 27.11.2020

- Die Übergabe der Zwischenablage an Windows funktionierte nicht, wenn kein scrp_write ausgeführt wurde.
- Neue Empfehlung win_lnk nicht über den Auto-Ordner (autoexec.bat) zu starten sondern per Anwendung anmelden
- Wenn man win_lnk per Hand oder per Anwendung anmelden startet kann man auch wieder die Definition der Windows-Endungen per Environmentvariable nutzen.
- win_lnk kann jetzt auch formatierten Text (rtf) an Windows weitergeben. Die andere Richtung klappt nicht, da dies von MagicPC ausgeführt werden müßte. Dies ist im ß-Stadium

Dimitri Junker