

in

COLLABORATORS

	<i>TITLE :</i> in		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	in	1
1.1	main	1
1.2	add8svx	2
1.3	comline	2
1.4	fsearch	3
1.5	grx	4
1.6	guide	4
1.7	ilbmsize	5
1.8	movescreen	5
1.9	nocr	5
1.10	nodes	6
1.11	proptime	6
1.12	rep	7
1.13	small	8
1.14	wbscreen	8
1.15	wrap	8
1.16	index	9

Chapter 1

in

1.1 main

Guide for the commands in GUIs:C

=====

These commands are used in various guis contained in the Gui4Cli archive. They are all pure and can be made resident. They are CLI commands - do not run them from the Workbench. Although they were written for use with Gui4Cli, they can be used anywhere else too (like DirOpus, DM etc).

CLI Commands :

=====

FSearch

Lots of options - read them..

Rep

FromFile/A,String/A,ToString/A,ToFile,I=Insensitive/S

Wrap

FILE/A,NEWFILE,L=LENGTH/K/N,TAB/K/N,P=PARA/K/N,STRIPCR/S,ADDCR/S

Grx

RexxPortName CommandLine

GetNodes

GUIDE/A,FILE/A

WriteNode

GUIDE/A,START/N/A,LENGTH/N/A

SplitGuide

GUIDE/A,DIR

JoinGuide

SOURCE/A,GUIDE/A

```
Add8SVX
FILE1/A, FILE2/A

ILBMSize
ILBM_FILE/A

ProgTime
<command line>

MoveScreen
SCREEN/A, DISTANCE/N/A, STEP

WBScreen
OPEN/S, CLOSE/S

NoCR
FILE/A

ComLine
<command line>
```

These programs are FreeWare, but remain the Copyright of D.Keletsekis.
No guarantees as to their performance or actions is made or implied.
Use them at your own risk.

D. Keletsekis
dck@hol.gr

1.2 add8svx

```
ADD8SVX File1/A, File2/A
```

Will APPEND (join) 2 8SVX sound samples.

The 1st file given will be OVERWRITEN and will include both
samples - i.e. File2 will be APPENDED to File1

So make sure you have a backup copy of File1..

1.3 comline

```
ComLine <command line>
```

This may well be the most useless command in existence..

All it does is repeat the command line you throw at it.

It's reason for existing is for testing command lines.
(look at `guis:tools/rtn/FindFile`)

1.4 fsearch

FSearch is a pure CLI command for Listing & Searching files.

Searching of files is fast and the template provided allows you to accurately describe the types of files you are looking for, thereby drastically reducing the scope of the search.

Fsearch has the following template :

ROOT/A/M	Where to start searching from. May be multiple root sources - with wild-cards ex : FSearch ram: dh0:mydir#? all List all files in all dirs in ram: and all files in all dirs matching dh0:mydir#?
	This option must be given. All others are optional.
PAT/K	(string) File pattern to match Only files meeting this pattern will be searched ex : FSearch ram: pat=#?.info all List all info files in ram: or it's sub-dirs
TXT/K	(string) Text to search for. By default the search is case-insensitive. If this option is not specified FSearch will act like "list". ex : FSearch ram: txt="my name" all
FROM/K	(date) Only search files FROM this date and up. Dates must be written as dd-mm-yy. ex : FSearch ram: from=07-08-96
TO/K	(date) Only search files UP to this date
MIN/N/K	(number) Only search files more than MIN Kb. ex : FSearch ram: min=100 (look for files >= 100k)
MAX/N/K	(number) Only search files less than MAX Kb.
HEADER/K	(string) Only search files which have this header. The Header specification may contain wild cards. The first 100 bytes of the file will be loaded and checked against this Header. ex : FSearch ram: header=FORM????ILBM#? all will list all iff pictures in ram:
ALL/S	Recursively scan sub-directories

CS=CASESENSITIVE/S
 Make search case-sensitive

V=VERBOSE/S
 Print Line number & text of lines found.
 The default is to just print the file name on the
 first occurrence of the TXT you're looking for.
 With this option all the matches and their
 context will be displayed. If the line length is
 over 80 chars it will be clipped.

NOBIN/S
 Do not check binary files. Up to 100 bytes of the
 file will be loaded and file will be skipped if
 this header contains any non-ascii characters.

HL=HIGHLIGHT/S
 Highlight the search results

INFO/S
 Print file size/date/time next to it's name
 No file paths are printed if this option is on.

NOPATH/S
 Print only the file name - not the full path.

1.5 grx

GRX PORT/A, COMMAND/F/A

The purpose of this command is to send ARexx commands to the given PORT.

Example : GRX Gui4Cli GuiLoad Guis:Demo.gc

I wrote this for Gui4Cli but you may use it for any program which has an ARexx port. Just state the PORT and the COMMAND you want to send.

RexxMast need not be running.

1.6 guide

SplitGuide GUIDE/A,DIR
JoinGuide SOURCE/A,GUIDE/A

These 2 commands were used in making the Gui4Cli guide.

Splitguide - will split an AmigaGuide document into files, each node becoming a file with the same name as the node.
GUIDE - is the guide name
DIR - (optional) is the dir to place the resulting files.
if DIR is not declared, they will be placed in ram:

JoinGuide - will join all the files in a directory into a complete AmigaGuide document, also making an INDEX of the nodes.
SOURCE - is the name of the DIR where the files (nodes) are
GUIDE - is the name of the resulting guide.

I made these because I find it a lot easier to work with small files than with a huge 200k bugger..

1.7 ilbmsize

```
ILBMSize ILBM_FILE
```

A simple command which will print out the size & depth of an IFF-ILBM picture.

The output will be something like :

```
> 400 200 5
```

meaning : width=400 pixels, height=200 pixels, depth=5 bitplanes

You can get the number of colours by calculating 2 to the power of the number of bitplanes, i.e. in the above example, the picture has 2 to the 5th => $(2*2*2*2*2) = 32$ colours

May come useful in scripts..

1.8 movescreen

```
MoveScreen SCREEN/A,DISTANCE/N/A,STEP
```

This is a little command to move a screen up or down.
This command will not change the world as we know it, but still..

```
SCREEN    is the name of the screen that gets moved  
DISTANCE  is the number of pixels to move it (+ or -)  
STEP      (optional) is the pixels to move each time
```

1.9 nocr

```
NoCR <FileName>
```

Remove all carriage returns which PC texts have.

1.10 nodes

GetNodes and WriteNode commands

These programs are meant to be used with the Gui4Cli help system. They will act on AmigaGuide files.

GetNodes :

GetNodes will extract all the nodes an AmigaGuide document has and save them in a list with the OffSet into the file and the Length of each node.

Template :

GetNodes GUIDE/A,FILE/A

GUIDE the name of the guide

FILE the name of the file to save the nodes list in.

WriteNode :

WriteNode will extract the text of the given node and save it in a file RAM:GUIDE.txt. From there, it can be read into a G4C lister. START and LENGTH are the values saved out by GetNodes.

Template :

WriteNode GUIDE/A,START/N/A,LENGTH/N/A

GUIDE the name of the guide

START where the node starts

LENGTH length of node text

The text will be formated in a non-AmigaGuide format.

1.11 proptime

ProgTime <command line>

- measures how fast a CLI command executes

Use : ProgTime <command line>

ex : Proptime Fsearch MyFile SomeString

Result : PROGRAM TIME : 0 minutes, 4 seconds, 15 ticks

1.12 rep

REP is a small, fast CLI command for replacing strings in files. It is pure and can be made resident.

This version adds the ability to define HEX and non-printable characters in the string (read on..)

Usage :

```
> Rep FromFile/A, String/A, ToString/A, ToFile, I=Insensitive/S
```

Where :

```
FromFile      - is the file you want to convert.
String        - The string you want to replace
ToString      - What you want to replace it with
ToFile        - (Optional) the name of the output file. If this is
                not given, then the INPUT FILE WILL BE OVERWRITEN!
Insensitive   - (Optional) Make the search case-insensitive
```

Example :

```
rep df0:MyFile ThisString ThatString Insensitive
- will replace all occurrences of "ThisString" (or THISstring, or
  ThIsStRiNg etc) with "ThatString" and save it over the original
  file.
```

"String" and "ToString" are parsed and may contain HEX, DECIMAL and other non-printable characters. The way to do this is:

The "\" character starts a character specification :

```
\n      - is the Newline character (decimal 10)
\r      - is the carriage return
\t      - is the TAB character
\       - is the \ character itself
\"      - is the " character
\'      - is the ' character

\x      - starts a HEX character string, defined by pairs
          of HEX digits (0-9,A-F) until another \ character
          or the end of the string are encountered.
          ex: MyString\x000003F3\continues

\#      - starts a DECIMAL character, defined by a number
          from 0-255. Again, the number ends with another
          \ character or the end of the string.
          ex: "\#155\32mWhite text"
```

Any other character will be considered literally, i.e.

```
\c      - is the "c" character.
```

Note that some of this parsing may conflict with Gui4Cli parsing - you may have to consider the fact that Gui4Cli will translate the line first, *before* rep gets a chance to

parse it as above..

1.13 small

ComLine <command line>

This may well be the most useless command in existence..
All it does is repeat the command line you throw at it.

It's reason for existing is for testing command lines.
(look at `guis:tools/rtn/FindFile`)

NoCR <FileName>

Remove all carriage returns which PC texts have.

1.14 wbscreen

WBScreen OPEN/S,CLOSE/S

This command will open or close the WB screen (no extra points for having already guessed that..)

WB will be closed **if** possible - i.e. if all windows are closed.
Otherwise a return code of 5 (WARN) will be returned.

I should really add this fuction to Gui4Cli but..

1.15 wrap

Wrap FILE/A,NEWFILE,L=LENGTH/K/N,TAB/K/N,P=PARA/K/N,STRIPCR/S,ADDCR/S

- ReWraps a text file - many options

FILE = Must be given. It's the file which you want to rewrap and it must be a text file!

WARNING - if you re-wrap a binary or a PowerPacked or XPK etc file, the file will become trash!

NEWFILE = This is the name of the output file. If it is not given, then the input FILE will be overwritten!

LENGTH = The new line length you want. If this is not given, the default is 1000 characters, which results in leaving

the line length same (unless you have lines over 1000 characters long). If the Length given is too small to fit a single word, then the word will be split. Max length is 1000 characters.

TAB = If this option is given, then any tabs found will be dealt with in accordance with the number of spaces specified, and be converted to spaces. Maximum tab size is 80 characters.

Warning If TAB size is not declared, all tabs will be deleted. This is not as bad as you may imagine, since Re-wrapping files with a lot of tabs looks ugly anyway..

PARA = This option is weird but usefull. It lets you define the number of NEWLINE characters that Wrap will read before putting 1 NEWLINE character. Most text documents have a NEWLINE character at the end of every line, and 2 NEWLINES to denote a paragraph.

So if you put PARA=2 you will get much better formatted text because the single NEWLINE characters will be ignored. This option also allows you to re-wrap files to a longer line length than the original file.

The default is PARA=1, i.e. leave them un-altered. Maximum is 30 newline characters.

STRIPCR = Strips those PC Carriage Return charaters.

ADDCR = Adds Carriage returns after every NEWLINE character, so you can tell PC users what you really think..

By declaring both STRIPCR *and* ADDCR you could ReWrap a PC file and keep it in PC format.

Example :

Wrap MyFile MyOutputFile LENGTH=60 TAB=5 PARA=2 STRIPCR

1.16 index

Guide INDEX :

Add8SVX

Comline

FSearch

grx

Guide

ILBMSize

MoveScreen

NoCR

Nodes

ProgTime

Rep

Small

WBScreen

Wrap
