

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	cli.gc	2
1.3	filepop	2
1.4	flereq.gc	3
1.5	findfile	3
1.6	getfiletype	4
1.7	getstring	5
1.8	palette.gc	5
1.9	read.gc	6
1.10	viewfile	6
1.11	index	7

Chapter 1

in

1.1 main

* Gui4Cli Helper Guis *

This is a description of some of the more interesting helper guis contained in the Gui4Cli archive, which you can use to enhance your own guis.

```
Cli.gc
CommandLine Mode

FilePop
FileName

FileReq.gc
GuiName Routine Path FavFile

FindFile
FileName

GetFileType
FileName

GetString
Title String Gui Routine Multi

Palette.gc

Read.gc
FileName

ViewFile
FileName FileType Mode
```

dck@hol.gr

1.2 cli.gc

CLI.gc CommandLine Mode

PURPOSE : Open a gui like the WB "Execute Command.." gui, but more.

PATH : guis:tools/cli.gc

USE : GuiLoad guis:tools/cli.gc CommandLine Mode

where :

CommandLine is the command line you want (can be "")

Mode (Optional) is one of RUN, CLI, AREXX, EXEC

Use GUILOAD to open the gui even if it's already loaded.

The gui will open with "CommandLine" in the textin gadget, and the cyclar set to "Mode". Thereafter the user can use it as needed.

1.3 filepop

FilePop FileName

PURPOSE: A small pop-up gui to deal with a file.

PATH : Guis:tools/rtn/FilePop

USE : GUILOAD Guis:tools/FilePop FileName
or..
GUIOPEN FilePop FileName

This gui is used in FSearch.gc and others.

You just give it a file name and let it do it's stuff..

It will pop-up a small window with 5-6 buttons to let the user decide what action to take with a file.

The 1st button is "AUTO" which checks the file type and runs it accordingly, using the \$DEF.PLAYERS defined.
(see the guis:tools/config.gc gui)

The 2nd button is "MORE..." which opens an other gui FILEPOP2 which has more choices. These can be changed/extended as needed.

After use the gui will remain in memory for faster loading if you want to use it some more. You have to specifically GuiQuit FILEPOP to unload it.

1.4 filereq.gc

FileReq.gc GuiName Routine Path FavFile

PURPOSE : A multiselect File Requester with Favourites etc.
Used as a replacement for the ASL requester

PATH : guis:tools/filereq.gc

USE : GuiLoad guis:tools/filereq.gc GuiName Routine Path FavFile
or..
GuiOpen filereq.gc GuiName Routine Path FavFile

where..

GuiName - is the name of your gui

Routine - is the name of a routine in your gui that will be
called repeatedly (with gosub) for every file chosen in
the requester. (see below)

Path (optional) the path you want the file requester to
open at - if you want..

FavFile (optional) the full path&name of the Favourites file
you want to use - default is guis:tools/fav/default

NOTE: if you want to give a FavFile but not a Path, you must
still state the Path argument (using "")

The file requester will let you choose files (multiselect or double click)
and will send the files chosen back to a specified "Routine" which exists
in file "GuiName".

The "Routine" will be called as follows:

--> GoSub \$GuiName \$Routine FileName

i.e. you'll get the file name as the 1st argument.

If the user selected many files, then this routine will be called
repeatedly for every file.

It's very easy to use and has many goodies, such as definable Favourites
file, delete/copy capabilities etc.

Look at guis:g4c/ced/cedbar.gc for a working example.

1.5 findfile

FINDFILE FileName

PURPOSE : Ask the user to show us where a file is
PATH : guis:tools/rtn/FindFile

USE : GUILOAD Guis:tools/rtn/FindFile FileName

A gui to guide a user to finding a file - or a path to a certain program.

The argument passed must be the name of the file/program/anything you're looking for. For example :

```
GuiLoad guis:tools/rtn/FindFile "DPaintV"
```

RETURNS :

Upon it's returning, \$\$RET.0 either contain :

- The full path of the file, or
- Nothing (i.e. \$\$RET.0 = ""), or
- A dummy command guis:c/ComLine, which will just print out the command line you send to it. This is usefull for gui previews.

note : no window - Just a load routine which runs & quits

1.6 getfiletype

GetFileType FileName

PURPOSE: This routine tries to recognise the type of a given file.
PATH : guis:tools/rtn/GetFileType

USE:

GUILOAD guis:tools/rtn/GetFileType FileName - or
GUIOPEN GetFileType FileName

This routine tries to determine the file type of "FileName"

It will first look at the file header, then at the extension, until it determines the type.

When it returns, \$\$RET.0 will contain :

- The type of file that "FileName" was, or
- "NONE" if there was no file, or
- "UNKNOWN" if the file was not one of the various filetypes defined in the routine (to which you can easily add..)

The type of file is returned as a word (such as ILBM or GIF or 8SVX etc)

See the file itself to see the exact words.

This function is used in conjunction with `guis:tools/rtn/viewfile` to run any file according to it's type.

1.7 getstring

```
GetString Title String Gui Routine Multi
```

Get a string from user & pass it to \$Gui \$Routine given

Use :

```
GUILOAD GUIS:TOOLS/RTN/GETSTRING Title String Gui Routine Multi
```

where :

```
Title      is the window title you want
String     is the starting string you want in the xTEXTIN gadget
Gui        is the name of the gui calling the routine
Routine    is the routine to call
Multi      (optional) is a flag which if = "Multi" then gui will not quit
           This is used for multiple use
```

This gui is used in the `dir.gc` file manager for the `Rename`, `Makedir` and `Assign` functions - you can see it in `guis:dir/dir.g`

It opens a window with "title" and a textin gadget containing "String". The user will change the "String" shown in the textin gadget and when he presses <enter> it will call the given routine, passing the changed string as the 1st argument :

```
GoSub $Gui $Routine $NewString
```

The \$Routine called must be prepared to deal with the string.

If the "Multi" argument is declared, the gui will stay open and can be called repeatedly (with `GuiLoad`). Otherwise it will quit.

1.8 palette.gc

```
Palette.gc
```

```
PURPOSE : This is a palette gui, for setting/saving/loading palettes.
PATH    : guis:tools/palette.gc
```

```
USE     : GuiLoad Palette.gc (it will open on it's own)
```


You can use this as a palette gui anywhere.

- The Palettes are **not** saved as normal IFF palette definitions, but as GUIs.
- They are kept in drawer guis:tools/palette
- You can load them as follows :

```
GUILOAD guis:tools/PaletteName GuiName
```

where

PaletteName is the name of a previously saved palette
GuiName is the name of your gui (i.e. the gui whose screen colors will be changed).

1.9 read.gc

```
Read.gc FileName
```

PURPOSE : A text file Reader.
PATH : guis:tools/read.gc

USE:

- GUILOAD guis:tools/read.gc FileName

This is a window with a listview, to be used for reading text (or other) file. It has file search and edit capabilities.

All you do to use it is pass it the filename you want displayed. If no filename is given a requester will open.

1.10 viewfile

```
ViewFile FileName FileType Mode
```

PURPOSE: This routine will run "FileName" according to what "FileType" it is.

PATH : guis:tools/rtn/ViewFile

USE : GuiLoad guis:tools/rtn/viewfile FileName FileType Mode
or..
GuiOpen ViewFile FileName FileType Mode

where:

FileName is the name of the file you want to run
FileType is a word as returned from routine Guis:tools/rtn/GetFileType
Mode (optional) can be RUN or CLI (RUN is default)

What you do to use it, is the following:

Say you have a file called MyFile and you want to view it.

1. You must run it through the GetFileType routine to get the file type it is. GetFileType will return something like "ILBM" (if the file is an iff picture, for example)
2. You then call this routine.

The name of the players used to play/run/execute etc each type of file are defined as Global variables (ex: *DEF.GIF etc).

These Global Variables are gotten through the guis:tools/CONFIG.GC gui.

1.11 index

Guide INDEX :

Cli.gc

FilePop

FileReq.gc

FindFile

GetFileType

GetString

Palette.gc

Read.gc

ViewFile
