Injector.doc

COLLABORATORS			

	TITLE : Injector.doc		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		December 7, 2024	

REVISION HISTORY			

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Injector.doc

1.1 Injector documentation - English

Welcome to Injector 2.32 copyright 1993-94 Frédéric Delacroix.

This is the document that must be spread with all other files of the package. It is to be viewed by AmigaGuide (copyright Commodore), but can also be read by human eyes, with less comfort.

TABLE OF CONTENTS

1.2   1.1 Distribution and disclaimer

Injector is Copyright 1994 by Frédéric DELACROIX. Permission to copy & distribute it is granted to anyone who follows this conditions (generally known as SHAREWARE):

  - All files (or the whole archive) are distributed together (this
includes the executables, documentation files, catalog files, the catalog
transcription file and all the icons). There is one exception though: If
you're planning a release for a "mono-linguistic" community (say, for
example, France only), you are allowed to distribute only the files that
are relative to your language, that is to say: the doc file and the
catalog file. All other files must be present.
  - All distributed files remain UNMODIFIED IN ANY WAY (no silly
"spread by..." message). If you have comments to add, do so in a separate
file! Archiving is although allowed, but executable crunching is not
recommended as the program is self-detaching (and thus needs to cut its
SegList).
  - You may NOT ask for money for this program. A small fee for copy
& sending is allowed, but you may NOT charge more than Fred Fish does for a
single AmigaLib disk.

  - This is for all users of the Injector: as the program is
released as SHAREWARE, you must send me a little contribution of $10 (or
equivalent) if you continue to use it after a short evaluation period. If
you want the source of the program (written for Devpac 3), just add $10
more and I will send it to you. My address can be found at the end of the
document.

Injector was tested a lot, but I cannot guarantee that it will always work
as expected. I will not be held responsible for any direct or indirect
damage/ loss of data that might result by the use of this program.
Remember: you use it AT YOUR OWN RISK.

Special note: the catalogs and doc files are only available in english and
french for now (my school spanish is soooo bad :-). So if you can make a
translation of these files into your own language, I would appreciate you
send it to me so I can include it in a future release. For the doc file,
it's easy: just edit a copy of this one (please, don't change the node
names, they are not displayed anyway). For the catalogs, just fill in the
.ct files with your own strings and send it to me. I will send you
corresponding catalogs in return. Please note that some strings (like the
Commodities description message and gadget labels) are limited in length.
For some, there is a shortcut key (in upper case please) or menu equivalent
right before the label.
  I will not distribute the cd file with this program. If you want a
translation, send me the ct file along with you registration fee...

Injector has now received nicer icons. These were drawn by Christophe
Peugnet (from ProMedia) and are copyrighted by him and distributable only
with Injector. In the directory OtherIcons, there are two more icons,
drawn by Laurent Pazdziej, copyrighted by him and distributable only with
Injector.

Forgot to mention in ealier versions (I apologize...) that the icons I used
for the drawers and some files are from IconPack, which is a collection of
nice icons drawn and copyrighted by Tom Ekström.

1.3 1.2 Credits and special thanks

Injector was written on the wonderful HiSoft Devpac 3, running on an old
A500 with OS 2.1. It uses the great reqtools.library which is Copyright
Nico François and errormsg.library, which is copyright by myself. ARexx is
copyright ©1987 by William S. Hawes.

The concept of Injector was originally implemented in the FR_Bypass program
that was part of the kd_freq.library (Copyright Khalid Aldoseri)
distribution, but I have totally rewritten the program, and now it doesn't
need that library any more. You can still use a patch program like RTtoKD
to replace reqtools's FileRequester by kd_freq's.

Thanks to: Nico François for the reqtools.library
    Khalid Aldoseri for his kd_freq.library
    William S. Hawes for ARexx
    Hisoft for Devpac 3
    Commodore for the amiga and its OS
    CEI for saving the amiga (?)
    J.P.Deloffre and C.Peugnet at the ProMedia association
    Christophe Peugnet for his nice icons
    Laurent Pazdziej for more icons
    Yves Quiquempois for Enforcer testing and improvement ideas
    AmigaNews.

1.4   1.3 System requirements

Injector should work on any amiga that matches these requirements:

  – You need Kickstart 2.04 or newer (V37+), or Injector will refuse
to run. The only thing to do if you are still under 1.3 is UPGRADE !
Believe me, it's worth it.

  – You need the reqtools.library V38+ (release 2.1) installed in
your LIBS: directory. It is not provided in this archive but you can find
it nearly everywhere in public domain collections.

  – You need the errormsg.library V4.0+ installed in your LIBS:
directory. It is provided somewhere in this archive.

  – You need the iffparse.library V37+ installed in your LIBS:
directory. This came along with Workbench 2.0+. V39 is even better as bug
free (works with any OS!).

  – Injector uses the locale.library V38+ if it is available to
support multiple languages. This library is normally shipped with OS 2.1+.

  – The preferences program makes use of the amigaguide.library to
display this file whenever the Help key is pressed, but it is not
mandatory.

  – ARexx is needed for the ARexx facilities of Injector (!). You
can find ARexx along with the original distribution of Workbench 2.0 and
higher.

1.5   2.1 Introduction

Those of you who already know Injector will have to read on, as the program was entirely rewritten since version 1.x, and lots of things have changed.

Injector is a hotkey-invoked commodity whose purpose is to inject things into the InputEvent stream (that is to say as if typed on keyboard). It is wholly configurable with a nice preferences program, has ARexx facilities and lots of features.

However, the icon facility has been removed, this will be the job of a more functionnal commodity (yet to be written:-).

To make this work, Injector has its own language, made of keywords, with or without arguments enclosed in parentheses.   You will have to read the references sections for further informations.

1.6   2.2 Starting Injector

   Injector can be started either from CLI of from Workbench.

   If used from the CLI, Injector will automatically detach itself, enabling the CLI window to close or the startup-sequence to continue, no need for the Run command. Unlike previous versions of Injector, there are no CLI arguments recognized. The config file provides for that. Some more options are descibed in the tooltypes section.

   If you use Injector from the Workbench, the best place for it to be is the WBStartup drawer. This way, Injector will be started each time the system boots up (don't forget to register!).   You can of course double-click its icon too.

   Even when started from Workbench, Injector cuts its SegList so that Workbench can close while Injector is active. Plus, there is no need for the DONOTWAIT tooltype.

1.7   2.3 ToolTypes

   This section describes the tooltypes that are recognized by Injector.   Tooltypes are the ONLY way to pass these arguments, even from CLI. They are: PUBSCREEN,CX_PRIORITY,PREFSPATH, AREXXCONSOLE, BREAKKEY and TOOLPRI.

1.8   The PUBSCREEN tooltype

This tooltype tells Injector which public screen should be used for the file requester, the string requester, the public screen requester and the key requester.   The default is '*', which means that the frontmost screen will be used, provided it is public (just as CON: windows). If the screen is not available, the default public screen is used.

1.9   The CX_PRIORITY tooltype

This is used to specify a number that will be used as the priority of
Injector's Broker in commodities' list. This is useful if you want
Injector's hotkeys to override another commodity's hotkeys (or the
contrary). The higher the priority is, the sooner Injector will receive
the InputEvents (relatively to other brokers). The default is 0.

1.10   The PREFSPATH tooltype

This option will tell Injector where to find the preferences program
whenever it is invoked by the EXEC_PREFS command. The default is
'InjectorPrefs', which must of course be in your valid path. I do use
PREFSPATH=SYS:Prefs/Injector .

1.11   The AREXXCONSOLE tooltype

This tooltype describes the console window Injector will open when
executing an ARexx program. This MUST be an interactive console, for
example a CON: window (maybe an AUX: stream ? I haven't tried it).

The default is CON:////Injector and ARexx/SCREEN*/AUTO/WAIT/CLOSE, which
will provide you with a reasonable console window on the front screen if
public, or the default public screen.

1.12   The BREAKKEY tooltype

This tooltype is new as of version 2.28. It is a commodities description
string used to tell Injector which Hotkey should stop any action by
Injector.
This "break key" feature was introduced to prevent Injector from injecting
all sorts of things and go out of control. If for example, you have
selected the wrong file, you can still stop the injection with this key.

Contrary to "normal" hotkeys, the event is NOT removed from the chain.
This way you can quit Injector at the same time, or stop another program
like WindowKey at the same time. This could have disastrous effects if the
application owning the active window recognizes this message.

The default value is lcommand rcommand lshift rshift q

1.13   The TOOLPRI tooltype

This tooltype is only recognized as of version 2.30. It is a standard
tooltype, as understood by Workbench, to set the priority of Injector's
task. However, as Injector launches its own process, the priority was
always -1 in earlier versions. Now you can set it with this tooltype,
which is also recognized for CLI.

1.14 2.4 Removing Injector

Injector can be removed by several ways. You can of course use the Exchange program and select Kill Injector. You can also run Injector (the master program, not the preferences program) again. The last way of removing Injector is to make it execute the CMD_QUIT command.

In all cases, Injector cannot be removed while an ARexx program is still running, as Injector must wait for the reply message from ARexx (don't want to wake up the guru, do you ? :-). Future versions might implement a "delayed quit" feature (is it really useful?).

1.15 3.1 Starting the preferences program

To modify its configuration, Injector makes use of a separate program, called the preferences program. This way, memory is not used to store unused data while Injector is active: the preferences program is loaded only when necessary.

The preferences program can be started either from CLI, Workbench, or Injector. It ALWAYS recognizes these tooltypes (even from CLI):

PUBSCREEN=<public screen name>. This defines the screen that should be used to open the window. The default is *, meaning that the front screen should be used if it is public.

CREATEICONS=<YES or NO>. This tells the preferences program of the original state of the "Create icons ?" menu item.

ACTION=<USE,SAVE or EDIT>. This is valid only for project icons that have their default tool set to the prefs program, or given via multi-selection. Such project icons are configuration files that are to be loaded immediately by the prefs program. This ACTION tooltype tells the prefs program what to do with this congfiguration file.

USE will save the given file as the "current" configuration, that is to say into the file "ENV:Injector.Prefs".

SAVE will save the given file as the "permanent" configuration, that is to say into both "ENV:Injector.Prefs" and "ENVARC:Injector.Prefs".

EDIT is the default, it won't save anything (yet), but rather let you edit the file as if you had started the prefs program then selected the Open... menu item.

Moreover, there are some options that can be used on the command line, that OVERRIDE the above-mentionned tooltypes. The template is:
 FILE,PUBSCREEN/K,USE/S,SAVE/S,EDIT/S,NCI=NOCREATEICONS/S,CI=CREATEICONS/S

FILE is of course the the name of the file to load, PUBSCREEN has the same meaning as the tooltype of the same name, USE,SAVE and EDIT are switches that are equivalent to the corresponding values of the ACTION tooltype, and NOCREATEICONS is equivalent to the tooltype CREATEICONS=NO. CREATEICONS was

introduced as of version 2.10 and is equivalent to the tooltype
CREATEICONS=YES. The last two keywords may be used to override defaults, I
guess they won't be used very much.

1.16 3.2 The Graphics User Interface

When started in EDIT mode, the preferences program opens a window with some
gadgets and menues. If you're used to normal preferences programs, you
should not have problems using it. Yet here are some explanations.

The main display is a list. It contains all currently defined hotkeys.
You can select a name in that list by clicking on it, or using the shortcut
key (h for the english version) with or without SHIFT to scroll through the
list without using the mouse. As of V2.11, you can also use the up and
down arrows to move by one entry in the corresponding direction, by one
page when shifted, or go to the top or bottom of the list with control.

On the right, there are 5 gadgets that are used to manage the appearance of
the list. They do not change the way Injector will treat it (except maybe
that if two hotkeys conflict, the first one will be used). Top will send
the selected entry to the top of the list, Bottom will send it to the
bottom, Up will move it one entry upwards, Down one entry downwards, and
Sort will sort the list in alphabetical order.

Below are some control gadgets. There are three string gadgets that are
used to edit hotkeys. For injector, a hotkey is a combination of three
things: a name, a key descriptor, and a command line. The 3 string
gadgets let you edit those fields.

New as of version 2.10, there is a checkbox on the right of the key
descriptor gadget. If it is checked, then the hotkey is enabled (as was
the case before version 2.10). If it is not checked, then the hotkey will
not be available. However, the hotkey is still available via the
EXEC_NAMED() command.

The Name field is used for displaying the list and by the EXEC_NAMED()
command.

The Key string gadget lets you edit the field that will tell
commodities.library which key combination should trigger the action of
Injector. This must be a valid commodities descriptor string. See
Defining hotkeys for further information. After typing in that string
gadget, the preferences program asks commodities if the string is OK, and
will alert you if it is not. As of V2.10, if the description is correct,
then the checkbox is automatically checked, and unchecked if it is not.

The Command string gadget is there to hold the command stream Injector
should execute when receiving the event corresponding to that hotkey.
Injector uses its own "language" for defining such actions. See the
Language reference section for details.

On the right of the string gadgets are three boolean gadgets named Create,
Copy and Delete. They are used respectively for creating a new hotkey
(that appears at the bottom of the list), Copying an existing entry
(appearing right after the currently selected one) and Deleting an existing

Hotkey.

Farther below are 5 gadgets. They control the behaviour of the program.
Save will store the edited file as the permanent configuration. Use will
save it as the current configuration. Both of them will make the
preferences program quit (unless an error occurs while saving).

Test will change the current configuration but won't end the prefs program,
so that you can test if the newly created configuration really suit your
needs, and change it if it doesn't.

Help will display this file. You need amigaguide.library V34+ for that.

Cancel will revert all you have done. If you had changed the current
configuration (via test), then it will be put back as it was before the
program was called. Then the prefs program ends.

That's it for the gadgets. Let's throw an eye at the menues:
```
+-------+    +----+        +-------+
|Project|   |Edit|        |Options|
+-------+---------+ +----+-----------+ +-------+---------+
|Open...   AO| |Last saved AL| |Create icons ? AI|
|Merge... AM| |Undo all AU| +----------------+
|Save as... AA| |Clear settings    |
+================+ +----------------+
|About...    |
+================+
|Quit   AQ|
+----------------+
```

Open will load a new configuration file. The currently edited one will be
lost. Merge will load a new file and merge it with the configuration you
are currently editing. Save as will write the currently edited file to a
new file. For these three actions, you will get an ASL file requester.

About will inform you about the version of Injector and my address. Quit
will end the program (Warning: no "Cancel" nor "Undo" is performed, a
tested configuration will remain active).

Last saved will get the last saved version of the permanent configuration
file (the configuration will be loaded from ENVARC:Injector.Prefs). Undo
All will cancel all the changes you have done. Clear settings will of
course clear the file you are currently editing. Use with care !

At last, Create icons is an on/off menuitem that tells the preferences
program whether it should create an icon for a file that is saved to disk
by the "Save as..." function. The icon will be a project, whose image will
be gotten from ENV:sys/def_prefs.info (or the default project icon if
inexisting), the default tool is set to the prefs program, and ACTION=USE
will be set as a tooltype. This way, double-clicking on that icon will
change the current configuration of Injector without really entering the
prefs program (does not mean it is not loaded though).

As of version 2.29, starting the prefs program again will activate, bring
to front and unzip it if necessary. Sending it a CTRL-F has the same
effect. Sending a CTRL-C will make it quit.

As of version 2.32, the prefs program is font-sensitive. The checkbox
might look strange with a font other than topaz/8 under OS2.0 because
gadtools does not scale it to the size of the gadget automatically. This
was fixed under OS3.0.

1.17   3.3 Language reference

This section will teach you how to write commands for Injector. These must
be entered in the Command string gadget of the preferences program's window
when a hotkey is selected, or sent via ARexx to the port named Injector.

A command is made of a keyword and an optionnal argument. Commands that
take arguments must have their keyword followed immediately by the argument
enclosed in parentheses. Thus it may be necessary for ARexx programs to
surround the parentheses by quotes, as the former are treated in a special
way by ARexx. But the parentheses must remain. Check the example programs
provided. Several commands are separated by spaces.

Now here is the list of the commands Injector understands (those that take
an argument have () after them):

```
  FREQ_SHOW    TIME_INJDATE()
  FREQ_INJFILE    TIME_INJTIME()
  FREQ_INJDIR   TIME_INJDAY()
  FREQ_INJPATH    TIME_LOCALIZED()

  SREQ_SHOW    EXEC_NAMED()
  SREQ_INJECT   EXEC_REXX()
        EXEC_PREFS
  CHARS_INJ()
  CHARS_CLIP()    CMD_QUIT
  CHARS_ENV()    CMD_UPDATE
  CHARS_RAWKEY()  CMD_NEWPREFS
  CHARS_FILE()
        PUB_SHOWLIST
  KREQ_SHOW    PUB_INJECT
  KREQ_INJECT    PUB_FRONT
```


1.18   The FREQ_SHOW command

This command will show the File requester. The user (you) can then select
a file, which can be pasted later with FREQ_INJFILE, FREQ_INJDIR or
FREQ_INJPATH.

The screen the file requester appears on can be set by the PUBSCREEN
tooltype.

1.19   The FREQ_INJFILE command

This command will inject the file name from a previously called file
requester (by FREQ_SHOW) into the InputEvent stream. Only the last
component of the name is pasted.

1.20 The FREQ_INJDIR command

This command is used to inject the directory name of a previously called
file requester. It works as FREQ_INJFILE.

1.21 The FREQ_INJPATH command

This command is like a medley of FREQ_INJDIR and FREQ_INJFILE, but it also
takes care of the handling of colons and slashes between the two. It
proves to be useful when using shell, for example.

1.22 The SREQ_SHOW command

This command will make Injector display the string requester. In this
requester, you can enter any string you want, limited to 80 characters in
length.

1.23 The SREQ_INJECT command

This command will inject the contents of the string requester, assuming it
was previously called by SREQ_SHOW. This is useful if you have to type
many times the same thing. As of V2.20, you can use escaped characters in
the requester, like newlines (\n), bells (\a) etc... The list of these
special characters is listed under the CHARS_INJ() command.

1.24 The CHARS_INJ() command

This command injects the constant string, which is given in argument, into
the InputEvent stream. Some special characters are recognized for this
command (they are the same that CatComp uses):
  \a Bell (ASCII 7)
  \b Backspace (ASCII 8)
  \c Control sequence introducer (CSI, ASCII 155)
  \e Escape (ASCII 27)
  \f Form feed (ASCII 12)
  \n New Line (ASCII 10)
  \r Return (ASCII 13)
  \t Horizontal tabulator (ASCII 9)
  \v Vertical tabulator (ASCII 11)
  \xNN Character with NN as hexadecimal ASCII code

```
\NNN  Character with NNN as octal ASCII code
\ Backslash
\) Close parenthesis (not end of arguments)
```

All are case-sensitive.

1.25 The CHARS_CLIP() command

This command will inject the contents of the system clipboard. To be precise, all CHRS chunks found will have their contents injected into the InputEvent stream, thus permitting an easy way of sharing data between applications (you can cut some text in the shell and inject it in a word processor that does not support the clipboard).

The argument is the clipboard unit number, from 0 to 255. This is generally 0 (for the shell etc...), but not necessarily.

1.26 The CHARS_ENV() command

This command will inject the contents of the environment variable whose name is given as an argument. Injector does no sanity checks, so avoid binary files !

1.27 The TIME_INJDATE() command

This command will inject the date according to the format requested by the argument.

If locale.library could not be successfully opened, the argument is forced to 2. Else, here are the possible values (other values are reserved for future expansion):
  0: short format, according to the current locale
     (ex: 20/04/94)

  1: long format, according to the current locale
     (ex: Mercredi 20 Avril 1994)

  2: DOS format. This is how the DOS displays the date.
     (ex: 20-Avr-94)

1.28 The TIME_INJTIME() command

This will enable you to paste the current time. The argument is a format number similar to that taken by TIME_INJDATE():

ex: 0: 18h46
  1: 18h47

```
  2:  18:47:10
```
(the french locale does not make any difference between long and short formats for the time).

1.29   The TIME_INJDAY() command

This is the final TIME_ command: it pastes the name of the day. The argument is still a format number (see TIME_INJDATE()).

```
ex: 0:  Mer
  1:  Mercredi
  2:  Mercredi
```


1.30   The TIME_LOCALIZED() command

This command is new as of version 2.03 of Injector. For it to work, you must have locale.library installed in your system. No injection will be performed if you do not have it.

This command is similar to other TIME_ commands; it enables you to inject the time with a custom display, according to locale.library's FormatDate() function. The argument is a string which will be pasted possibly including some formatting codes that will be replaced according to the following rules. All formatting codes all begin with a % sign, use %% for a real %. Those of you who know C's printf() function (or Exec's RawDoFmt()), won't be surprised. However, the codes are very different:

```
    %a - abbreviated weekday name
    %A - weekday name
    %b - abbreviated month name
    %B - month name
    %c - same as "%a %b %d %H:%M:%S %Y"
    %C - same as "%a %b %e %T %Z %Y"
    %d - day number with leading 0s
    %D - same as "%m/%d/%y"
    %e - day number with leading spaces
    %h - abbreviated month name
    %H - hour using 24-hour style with leading 0s
    %I - hour using 12-hour style with leading 0s
    %j - julian date
    %m - month number with leading 0s
    %M - the number of minutes with leading 0s
    %n - insert a linefeed
    %p - AM or PM strings
    %q - hour using 24-hour style
    %Q - hour using 12-hour style
    %r - same as "%I:%M:%S %p"
    %R - same as "%H:%M"
    %S - number of seconds with leadings 0s
    %t - insert a tab character
    %T - same as "%H:%M:%S"
    %U - week number, taking Sunday as first day of week
```

```
   %w - weekday number
   %W - week number, taking Monday as first day of week
   %x - same as "%m/%d/%y"
   %X - same as "%H:%M:%S"
   %y - year using two digits with leading 0s
   %Y - year using four digits with leading 0s
```

  So, for example, the command

TIME_LOCALIZED(Today is %A %e %B %Y, it is %H o'clock.)

  will inject the following (well, at the time I type it :-)

Today is Mardi 21 Juin 1994, it is 16 o'clock.

1.31   The EXEC_NAMED() command

This command will execute a hotkey as if the corresponding key had been
pressed by the user.   This way you can do GOSUB-like processings of
hotkeys. The argument is of course the name of the hotkey to be processed.
There is a security check in Injector: when a hotkey is executing, a
special bit in its structure is set, preventing recursive calls to be made
(you will just get a message).

1.32   The EXEC_REXX() command

This command tells Injector to launch the ARexx program whose name is in
argument.   The program will have its default host address set to
"Injector". The default file extension for such programs is .ijctr .

Thanks to this command (and ARexx!), you can generate complex macros, with
tests in them, etc etc etc...

1.33   The EXEC_PREFS command

This command will ask Injector to run the preferences program. This is
done via the SystemTagList() call of the dos.library, so the preferences
program must be in your path.   The default file name for it is
"InjectorPrefs", but it can be changed by the PREFSPATH tooltype.

As of version 2.11, this can also be acomplished by selecting Show in the
commodities exchange program. As of version 2.29, selecting Hide will end
the prefs program (by sending it a CTRL-C).

1.34   The CMD_QUIT command

The use of this command is obvious: on receiving it, and provided doing so is possible, Injector will suicide. This is not possible for example if some ARexx programs are still running.

1.35 The CMD_UPDATE command

This command is not useful in most cases. On receiving it, Injector will reload its configuration. This is useless in most cases since, like the IPrefs program, Injector uses the notification facilities of AmigaDOS to automatically detect alterations of the preferences file (by the preferences program for example). Generally, ENV: is assigned to the Ram disk, so it is not a problem. But all handlers do not support notification, so, if someone uses an assign over a network filesystem for example, he should use this command for Injector to update the configuration.

When this command is received, Injector does not continue the processing of the current line, even if there were commands remaining.

1.36 The CMD_NEWPREFS() command

This command will switch to another configuration file. The argument must be a valid path to the file in question. You can create such files by selecting the Save as... menuitem in the configuration program.

Injector makes the switch by invoking the preferences program with the argument and the USE keyword, so the prefs program must be accessible. You can use the PREFSPATH option for that.

This command is new for version 2.02 of Injector, usable since version 2.03.

1.37 The KREQ_SHOW command

This command opens the key requester window. This window will open on the screen specified by the PUBSCREEN tooltype, its width will be the eighth of the total screen width, so will be its height, and will be positionned right under the mouse pointer. This window waits for you to produce an event, like inserting a disk, removing a disk, pressing a (qualified) key or mouse button, inactivating the window.

In order to be able to receive truly qualified events, qualifier keys (shift, alt, control, etc) are filtered out (they do not produce rawkey events), and so are the mouse buttons when pressed (they do produce a rawmouse event when they are released). Apparently, Intuition does not seem to send a qualifier mask for IDCMP_DISKINSERTED or IDCMP_DISKREMOVED classes so qualifiers will be ignored when such events occur (not my fault). Moreover, as the input is done through Intuition, all keys used by InputHandlers with a higher priority (like commodities brokers) will not be

passed to the window, so such events cannot be tracked down.

Inactivating the window is considered by Injector as a cancel request. This can be done by clicking with the left mouse button outside the window. That's why when you want to produce left button events (as a qualifier or rawmouse event), you must click inside the window. As of V2.15, there is also a timeout of approximately 10s.

After having satisfied the window, you can translate the event into a commodities description string (see appendix 4.1) with the KREQ_INJECT command. This command is new as of V2.12 of Injector.

1.38   The KREQ_INJECT command

This command is to be used in conjunction with KREQ_SHOW: it injects the commodities description string for the event you gave to it. This is useful for defining hotkeys to be used by toolmanager, a blanker, and, of course, the preferences program of Injector.

Some features are unavailable (like the repeat qualifier, obviously), but you can still edit the produced string...

This command is new as of V2.12 of Injector.

1.39   The PUB_SHOWLIST command

This command, new for version 2.17, is used to pop up a little window with a list of all currently opened public screens. You can select one by clicking on a name in the list and selecting chose, or double-clicking a name. The list can also be controlled by the keyboard (arrow keys, RETURN for choose, ESC for cancel). The gadgets labelled Choose and Cancel are quite self-explanatory. The gadget labelled Front is used to automatically select in the list the name of the frontmost screen if it is public.

The window will remember its position after closing, and will select the name in the list that is the same as the one in the buffer (entered by PUB_SHOWLIST or, for version 2.18, by PUB_FRONT).

After having selected a name in the list, you can inject it with the PUB_INJECT command.

As of V2.32, this requester is font-sensitive.

1.40   The PUB_INJECT command

This command will inject a public screen name that was previously chosen with PUB_SHOWLIST or PUB_FRONT. This command is new for V2.17.

Example, this hotkey could be quite useful in a shell:

```
 CHARS_INJ(AmigaGuide Help:Autodocs PUBSCREEN=) PUB_FRONT
           PUB_INJECT CHARS_INJ(\r)
```

(see the provided configuration file).

1.41   The PUB_FRONT command

This command is new for V2.18. Its job is to fill the public screen name
buffer with the name of the frontmost public screen. If the frontmost
screen is not public, then no change is made to the buffer. This command
acts as if you had used PUB_SHOWLIST and selected "Front" and "Choose".
But no list is displayed. You can inject the result with PUB_INJECT.

1.42   The CHARS_RAWKEY() command

This command is new for version 2.23. The argument taken is a commodities
description string (as described in appendix 4.1). It is used to describe
the key to inject, be it a printable or non printable character, or even no
character at all, that is to say that strings like "rawkey lcommand
rcommand control lshift rshift numpad rbutton lbutton mbutton *" are
perfectly valid (although not really useable :-).   You may even use
rawmouse events.

As the events are injected with the input.device's IND_WRITEEVENT command,
such keys are propagated to all handlers, including, and this is where
CHARS_RAWKEY() becomes useful, to commodities brokers. You can thus define
synonyms for hotkeys or cumulate many actions on one hotkey. Personally, I
use the command named "Toolmanager docks" in the provided example
configuration file to open all my toolmanager docks at a time.

1.43   The CHARS_FILE() command

This command is new for version 2.27.   It enables you to inject the
contents of a file whose name is given in argument. Warning, no check is
done, so avoid binary files !

A special name is known: * means use the file that was chosen in a
previously invoked file requester by FREQ_SHOW.

1.44   3.4 Preferences versions

For V2.10,I had to include new features into the preferences file, meaning
2 extra bytes of data per hotkey. That's why the version number in the
preferences file has increased, and that's why Injector cannot read older
files. However, the preferences program can read old files, so you can use
it to convert your pre-V2.10 preferences files into new ones by loading
them in the prefs program and saving them again.

1.45   4.1 Defining hotkeys

The text that follows does depend on the behaviour of the commodities.library rather than on Injector's. This is how commodities will understand the key combination you want an action associated with. As of version 2.12, the KREQ_SHOW/KREQ_INJECT commands might also prove useful...

A description string is made as follows:

    [<class>] {[-][<qualifiers>]} [-][upstroke] [<key code>]
All keywords are case-insensitive.

class: set to an InputEvent class. Supported classes are rawkey for keyboard events, rawmouse for mouse events,diskinserted and diskremoved. The default is rawkey, which should generally be used.

qualifiers: this is a set of keywords representing the state of keyboard qualifiers (Shift,Alt etc...). This is a list of known keywords. Those marked with an * are new for commodities.library V38.
lshift,left_shift *:   Left shift key.
rshift,right_shift *:   Right shift key.
shift: Either shift key.
capslock,caps_lock *:   Caps Lock key.
caps: Either Caps Lock or shift.
control,ctrl *:   Control key.
lalt,left_alt *:   Left Alt key.
ralt,right_alt *:   Right Alt key.
alt: Either Alt key.
lcommand,lamiga *,left_amiga *,left_command *: Left amiga key.
rcommand,ramiga *,right_amiga *,right_command *:Right amiga key.
numericpad,numpad *,num_pad *,numeric_pad *: For numeric pad keys.
leftbutton,lbutton *,left_button *:   Left mouse button.(1)
midbutton,mbutton *,middlebutton *,middle_button *:
         Middle mouse button.(1)
rbutton:,rightbutton *,right_button *:   Right mouse button.(1)
repeat: Key-repeat active.(2)

Notes: (1) commodities.library V37 has a bug which prevents the use of
       leftbutton,midbutton and rbutton as qualifiers. It was fixed
       in V38.
    (2) for rawkey class only.
    (3) if a hotkey is to be insensitive to the state of a qualifier,
       place a - before the qualifier name.

upstroke: Normally an event is generated only when the key is pressed. You can change this behavior by setting upstroke. This will generate an event only when the key is released. If both press and release are to generate an event, use -upstroke.

key code: These are meaningful only to rawkey and rawmouse classes.
For rawkey, here are the key codes, * are new for V38 of commodities.
a to z, 0 to 9:   Normal ASCII characters.
f1 to f10,f11 and f12:   Function keys.
up,cursor_up *:   Up arrow key.
down,cursor_down *:   Down arrow key.

```
left,cursor_left *:     Left arrow key.
right,cursor_right *:     right arrow key.
esc,escape *:       Esc key.
backspace       Backspace key.
del        Del key.
help        Help key.
tab        Tab key.
comma        Comma key (,).
return        Return key.
space,spacebar *       Space bar.
enter        Enter key.(4)
insert *        Keypad 0 key.(4)
delete *        Keypad 1 key.(4)
page_up *        Keypad 9 key.(4)
page_down *       Keypad 3 key.(4)
home *        Keypad 7 key.(4)
end *        Keypad 1 key.(4)

Notes:  (4) must be used with the numericpad qualifier.

For rawmouse,valid keycodes are: (only valid for V38 of Commodities):
mouse_leftpress:       Left button pressed.(5)
mouse_middlepress:       Middle button pressed.(5)
mouse_rightpress:       Right button pressed.(5)

Notes:  (5) you must also set the corresponding qualifier.
```

1.46 4.2 History

```
Revision V2.32
--------------
    Font-sensitivity at last ! Will use topaz/8 only if the
    screen font is proportional. This is for the prefs program
    and the pubscreen requester.

Revision V2.31
--------------
    Adapted to version 3.02 of errormsg.library, causing execu-
    tables to shorten, messages to be removed. I will distribute
    only the library file itself and the installation script to
    reduce archive size.

Revision V2.30
--------------
    Now recognizes the TOOLPRI tooltype (as Workbench does).

Revision V2.29
--------------
    In the prefs program, tab activates the name string gadget.
    The prefs program will quit on receiving a CTRL-C, and the
    Hide commodities message is now recognized.

Revision V2.28
--------------
    Implemented the "break key" feature.
```

```
Revision V2.27
--------------
    Implemented CHARS_FILE.

Revision V2.26
--------------
    Bug fix: the key requester's injection would crash when the
    user used the space bar. (Bug reported by Y.Quiquempois).

Revision V2.25
--------------
    Added a zoom gadget for the prefs program.

Revision V2.24
--------------
    Fixed a minor bug which caused locale-based TIME_ commands
    to inject a nul character at the end of a string. Fixed a
    problem with the public screen requester.

Revision V2.23
--------------
    Implemented CHARS_RAWKEY.

Revision V2.22
--------------
    Injector still has problem to inject long strings on fast
    machines, that's why I now use input.device/IND_WRITEEVENTS
    instead of commodities.library/AddIEvents(). Hope it works
    now !

Revision V2.21
--------------
    Now uses DrawInfo pens for backfill pattern instead of sys-
    tematically using pens 0 and 2.

Revision V2.20
--------------
    You can now use backslashed characters (\n,\a etc) in the
    string requester. Some miscellaneous optimizations.

Revision V2.19
--------------
    Rewritten the character injection function: it now uses
    MapANSI() directly, instead of InvertKeyMap(). I hope it
    fixes a word-eating bug reported by friend of mine.

Revision V2.18
--------------
    Implemented PUB_FRONT.

Revision V2.17
--------------
    Implemented the pubscreen requester (commands PUB_SHOWLIST
    and PUB_INJECT).

Revision V2.16
```

```
--------------
     Added a nice backfill pattern for the key requester.
```

Revision V2.15

     Added a selected background color for string gadgets (in the
     prefs program) and a timeout for the key requester.

Revision V2.14

     Turned all move.l 4.w,a6 into move.l Exec.Base(pc),a6 ...
     I know I should have done this from the start, at last here
     it is.

Revision V2.13

     Optimized key requester window: size=size of screen/8, and
     position is under pointer. I have arranged the case of non
     printable characters: qualifiers other than shift keys are
     masked out during MapRawKey(), this should be fine for all
     currently used keymaps.

Revision V2.12

     Implemented the powerful functions KREQ_SHOW and KREQ_INJECT
     Still have to take care of non printable characters (like
     ctrl-w).

Revision V2.11

     Injector supports the SHOW message from commodities, it then
     calls the preferences program. The up/down arrows can now be
     used in the prefs program to scroll through the list.

Revision V2.10

     Hotkeys can now be disabled via a new checkbox in the GUI
     of the prefs program. This has caused a changed in the prefs
     file structure, but the prefs program can still read old
     files. Disabled hotkeys are still available using EXEC_NAMED
     Implemented the CREATEICONS CLI option.

Revision V2.09

     Fixed a Mungwall hit when selecting multiple menu items.

Revision V2.08

     Changed the default directory for Open, Save as and Merge
     from presets/ to PROGDIR:presets/ . This makes a lot more
     sense since the prefs program should be stored in the prefs
     directory.

Revision V2.07

     Well, this stupid bug is really strange. I have solved this
     crazy problem by opening amigaguide.library at startup time

rather than at Help-time. I also fixed the error report from
AmigaGuide (OpenAmigaGuideA() returns an AMIGAGUIDECONTEXT
even if an error occurs)...

Revision V2.06

    Bug fix: when the help file was not available and the user
    clicked twice (!) on the help button, the prefs program
    would crash. Thanks to Yves Quiquempois for reporting this
    strange bug.

Revision V2.05

    Fixed a little bug: if reqtools.library could not be opened,
    the iffparse.library was not closed.

Revision V2.04

    Reduced the size of the executable by removing some Reloc32
    and optimizing some code.

Revision V2.03

    Implemented the new TIME_LOCALIZED function, debugged
    CMD_NEWPREFS.

Revision V2.02

    Implemented the CMD_NEWPREFS function. Removed silly useless
    obsolete 8-byte part in the prefs program...

Revision V2.01

    Bug fixed: Injector would guru if no prefs file had been
    created... Sorry ! Minor changes have to be done very
    carefully !

Revision V2.00

    RUMBLE, RUMBLE... This is finally it (over 6 months after
    the first release!). I have totally rewritten the program,
    with lots of powerful features, a nice prefs program, full
    ARexx support, localization... It has little in common with
    the 1.x versions, so read the doc !

Revision V1.16

    Commodore have definitely gone wrong with their mis-
    documenting CreateNewProc(): the default value for
    NP_FreeSeglist is FALSE! This caused lots of memory wasted
    in earlier versions of Injector...

Revision V1.15

    Vicious bug fixed: Failed to unlock a public screens in some
    cases.

```
Revision V1.14
--------------
    Stupid bug fixed: PASTEFILEKEY and PASTEDIRKEY were inverted

Revision V1.13
--------------
    This is finally it! I've made it to the very first public
    version, released as SHAREWARE. As usual, I've done a bit of
    clean up and packed everything into a nice directory with
    nice readme files. (10-Oct-93  14:31:10)

Revision V1.12
--------------
    Added a 2.0-version string and written AmigaGuide doc files.
    No ASCII doc for this program,just get AmigaGuide (Fish 870)
    to view them. Optimized placement of icon window (right side
    of requester). Public release is for the next revision.

Revision V1.11
--------------
    The CXPRI tooltype was broken,and is now fixed. I have writ-
    ten a french catalog and I've fixed the locale interface.
    Done some miscellaneous bugfixes and optimizations. Planning
    a public release soon. Promised!

Revision V1.10
--------------
    Well, I have removed the automatic translation mechanism(was
    implemented with Commodities' Translate objects).Advantages:
    it shortens the program, commodities does not cause strange
    gurus when asynchronously accessing data at bad times. Draw-
    back: Cannot use hotkeys while requester is active.

Revision V1.09
--------------
    I've had the PubScreen and IconPath tooltypes working,and I
    have fixed a couple of bugs. Thinking of writing a doc file
    by now...

Revision V1.08
--------------
    I have made a skeleton for option support from CLI and tool-
    types. Supported options for now: all hotkeys,Commodities
    priority. Soon: IconPath and PubScreen.

Revision V1.07
--------------
    Wrote my own InvertString()/FreeIEvents() routines so that
    Injector does not misinterpret filenames with angled brakets
    any more. Also fixed self-detaching from Workbench to enable
    the use of the WBStartup drawer.

Revision V1.06
--------------
    Now the injector self-detaches from the CLI, no longer need
    to use Run. If the icon window is moved, the next time it
    opens at the same place.
```

```
Revision V1.05
--------------
    The displayed icon is finally a true selectable gadget,thus
    allowing you to see both inactive and active states. Note
    there could be a little difference with workbench's icons as
    Workbench has a "floodfill" mode which is not supported by
    standard Intuition gadgets. Might be fixed later.

Revision V1.04
--------------
    Ah, a cool new feature: the icon requester can now display
    the icons themselves in their own little window, using ano-
    ther nice ExtraButton. For now, the icon is not a true gad-
    get (hardly an image), but I shall fix it soon.

Revision V1.03
--------------
    That's it,Injector is now able to copy icons for directories
    ,just select an empty file name.

Revision V1.02
--------------
    Added interesting new feature: Injector can now copy icons
    from one place to another,using kd_freq's ExtraButton fea-
    ture. Proves to be very useful! Still have to take care of
    directories' icons.

Revision V1.00
--------------
    --- Initial release ---
```

1.47   4.3 Contacting the author

```
   I  remind  you that Injector is SHAREWARE.  I know most of you will
not pay me anything after their evaluation period, but please consider I've
spent  a lot of time writting and debugging this program, and I'd just like
a  small  reward that would enable me to buy a good hardware configuration.
Consider that paying $10 will provide you with support for future releases.
The  source  is  kept  warm  for  those  of  you who might be interested in
programming the wonderful amiga operating system, for $10 more.
   For anything like registration, comments, bug reports, improvements
requests, postcards, I can be reached at:

        Frédéric DELACROIX
        5 rue d'Artres
        59269 QUERENAING
        FRANCE.


Some things I have already thought of including in the next release:

- MUI version of the preferences program (if I manage to debug it !)
```