Injector.doc

COLLABORATORS			
	TITLE : Injector.doc		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		September 19, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Injector.doc

1.1 Documentation d'Injector - Français

```
                Bienvenue à Injector 2.00 copyright 1994 Frédéric Delacroix.

Ceci  est  le  document  qui  doit  toujours être distribué avec les autres
fichiers.   Il  est fait pour être lu par AmigaGuide (copyright Commodore),
mais peut aussi être vu par des yeux humains, le confort en moins.

                            TABLE DES MATIERES

1 AVANT-PROPOS:

                1.1 Distribution
                    <- Important !

                1.2 Crédits et remerciements

                1.3 System requirements
                2 INSTALLATION D'INJECTOR:

                2.1 Introduction

                2.2 Lancer Injector

                2.3 Les Tooltypes

                2.4 Terminer Injector
                3 CONFIGURER INJECTOR:

                3.1 Lancer le programme de préférences

                3.2 L'Interface Utilisateur Graphique

                3.3 Référence du langage
                4 APPENDICE

                4.1 Definir des touches de commandes

                4.2 Historique
```

1.2 1.1 Distribution et décharge

Injector est Copyright 1994 par Frédéric Delacroix. La permission de le copier et de le distribuer est accordée à quiconque respecte ces conditions (généralement connues sous le nom de SHAREWARE):

- Tous les fichiers (ou toute l'archive) soient distribués ensemble (cela concerne les exécutables, les fichiers de documentation, les fichiers catalogues, le fichier de transcription de catalogue et tous les icônes). Il y a une exception toutefois: si vous prévoyez une distribution pour une communauté "mono-linguistique" (disons, par exemple, la France uniquement), vous êtes autorisé à ne distribuer que les fichiers qui sont relatifs à votre langue, c'est-à-dire: le fichier de documentation et le fichier catalogue. Tous les autres fichiers doivent être présents.
- Tous les fichiers distribués ne soient MODIFIES EN AUCUNE FACON (pas de message idiot du genre "distribué par..."). Si vous avez des commentaires à ajouter, faites-le dans un fichier séparé! L'archivage est toutefois autorisé, mais le compactage de l'exécutable n'est pas recommandé car le programme se détache lui-même du CLI (il a besoin de couper sa SegList).
- Vous ne pouvez pas demander d'argent pour ce programme. Une petite somme est autorisée pour la copie et l'envoi, mais vous ne pouvez PAS demander plus que Fred Fish pour un AmigaLib disk unique.

- Ceci est pour tous les utilisateurs d'Injector: comme le programme est distribué en SHAREWARE, vous devez m'envoyer une petite contribution de $10 (50FF ou équivalent) si vous continuez à l'utiliser après une courte période d'évaluation. Si vous voulez le source du programme (écrit pour Devpac 3), ajoutez $10 de plus et je vous l'enverrai.

Mon adresse se trouve à la fin de ce document.

Injector a été beaucoup testé, mais je ne peux pas garantir qu'il marchera toujours comme prévu. Je ne peux pas être tenu pour responsable de dommages/ perte de données directs ou indirects qui pourraient resulter de l'utilisation de ce programme. Souvenez-vous que vous l'utilisez A VOS RISQUES ET PERILS.

Note particulière: les catalogues et les fichiers de documentation ne sont disponibles qu'en anglais et en français (mon espagnol scolaire est si mauvais :-). Si vous pouvez faire une traduction de ces fichiers dans votre langue, j'aprécierais grandement que vous me les envoyiez pour que je les inclue dans la prochaine version. Pour le fichier de documentation, c'est facile: éditez simplement une copie de celui-ci (merci de ne pas changer le nom des noeuds, ils ne sont jamais affichés de toute façon). Pour les catalogues, remplissez les fichiers .ct avec vos propres textes et envoyez-les moi. Je vous retournerai les catalogues correspondants. Notez que certaines chaînes (comme le message de description pour commodities et les étiquettes de gadgets) sont limités en longueur. Pour certaines, il y a un raccourci-clavier (en majuscules SVP) ou un équivalent à un menu juste avant l'étiquette. Vous pourriez aussi dessiner des icônes moins laids

pour Injector et le programme de préférences (je ne suis pas un graphiste :-).

1.3 1.2 Crédits et remerciements

Injector a été écrit avec le merveilleux Devpac 3 d'Hisoft sur un viel A500 avec l'OS 2.1. Il utilise la reqtools.library qui est Copyright Nico François et errormsg.library, qui est copyright par moi-même. ARexx est copyright ©1987 par William S. Hawes.

Le concept d'Injector été originellement implémenté dans le programme FR_Bypass qui faisait partie de la distribution de la kd_freq.library (Copyright Khalid Aldoseri), mais j'ai totalement ré-écrit le programme et maintenant il ne requiert plus cette bibliothèque. Vous pouvez toujours utiliser un patch comme RTtoKD pour remplacer le file requester de reqtools par celui de kd_freq.

1.4 1.3 Configuration minimale

Injector devrait marcher sur n'importe quel amiga qui remplit ces conditions:

- Il vous faut le Kickstart 2.04 ou plus (V37+), ou Injector refusera de fonctionner. La seule chose à faire si vous avez encore le 1.3 est de vous mettre à jour ! Croyez-moi, ça en vaut le coup.

- Il vous faut la reqtools.library V38+ (release 2.1) installée dans votre tiroir LIBS:. Elle n'est pas fournie dans cette archive mais vous pouvez la trouver presque partout dans les collections de domaine public.

- Il vous faut l'errormsg.library V1.0+ installée dans votre tiroir LIBS:. Elle est fournie quelque part dans cette archive.

- Il vous faut l'iffparse.library V37+ installée dans votre tiroir LIBS:. Elle est fournie avec le Workbench 2.0+.

- Injector utilise la locale.library V38+ si elle est disponible pour utiliser plusieurs langues. Cette bibliothèque est normalement fournie avec l'OS 2.1+.

- Le programme de préférences utilise l'amigaguide.library pour afficher ce fichier dès que la touche Help est pressée, mais elle n'est pas obligatoire.

- ARexx est nécessaire pour les options ARexx d'Injector (!). Vous pouvez trouver ARexx avec votre distribution originale du Workbench 2.0 et supérieur.

1.5 2.1 Introduction

Ceux d'entre vous qui connaissent déjà Injector devront continuer à lire, car le programme a été entièrement ré-écrit depuis les versions 1.x, et beaucoup de choses on changé.

Injector est une commodité invoquée par des touches de commande (à partir de maintenant appelées hotkeys) dont le but est d'injecter des choses dans la chaîne des InputEvents (c'est-à-dire comme si elles étaient tapées au clavier). Il est entièrement configurable avec un joli programme de préférences, a des options ARexx et plein d'autres choses.

Cependant, la copie d'icônes a été enlevée, ce sera le travail d'une commodité plus fonctionnelle (pas encore écrite:-).

Pour que cela marche, Injector a son propre langage, fait de mot-clés, avec ou sans arguments entre parenthèses. Vous devrez lire les sections de référence pour plus d'informations.

1.6 2.2 Lancer Injector

Injector peut être démarré à partir du CLI ou du Workbench.

S'il est utilisé à partir du CLI, Injector se détache automatiquement, autorisant la fênêtre du CLI à se fermer ou la startup-sequence de continuer, pas besoin d'utiliser la commande Run. Contrairement aux versions précédentes d'Injector, aucune option n'est reconnue sur la ligne de commande du CLI. C'est le rôle du fichier de configuration. Quelques options supplémentaires sont décrites dans la section

tooltypes

.

Si vous utilisez Injector à partir du Workbench, le meilleur endroit est le tiroir WBStartup. De cette façon, Injector sera lancé chaque fois que le système démarre. Vous pouvez bien sûr double-cliquer son icône aussi.

Même quand il est lancé du Workbench, Injector coupe sa SegList pour que le Workbench puisse se fermer quand Injector est actif. De plus, il n'y a pas besoin du tooltype DONOTWAIT.

1.7 2.3 Les tooltypes

Cette section décrit les tooltypes qui sont reconnus par ↩
Injector.

Les tooltypes sont la SEULE façon de passer ces arguments, même à partir du CLI. Ce sont:

PUBSCREEN

,

CX_PRIORITY

,

PREFSPATH

, et

AREXXCONSOLE
.

1.8 Le tooltype PUBSCREEN

Ce tooltype dit à Injector quel écran public doit être utilisé pour le file requester et le string requester. Par défaut, c'est '*', qui signifie que l'écran le plus en avant sera utilisé pourvu qu'il soit public (tout comme pour les fenêtre CON:). Si l'écran n'est pas disponible, l'écran public par défaut est utilisé.

1.9 Le tooltype CX_PRIORITY

Il est utilisé pour donner un nombre qui sera utilisé comme la priorité du Broker d'Injector dans la liste de Commodities. C'est utile si vous voulez que les hotkeys d'Injector surpassent celles d'une autre commodité (ou le contraire). Plus la priorité est haute, plus Injector reçoit les InputEvents tôt (relativement aux autres brokers). Par défaut, c'est 0.

1.10 Le tooltype PREFSPATH

Cette option dit à Injector où trouver le programme de préférences ↩
quand il
est invoqué par la commande
EXEC_PREFS
. Par défaut, c'est 'InjectorPrefs',
qui doit bien sûr être dans votre chemin d'accès. Personnellement, j'utilise PREFSPATH=SYS:Prefs/Injector .

1.11 Le tooltype AREXXCONSOLE

Ce tooltype décrit la fenêtre de console qu'Injector doit ouvrir quand il lance un programme ARexx. Ce DOIT être une console interactive, par exemple une fenêtre CON: (peut-être un canal AUX: ? Je n'ai pas essayé).

Par défaut, c'est CON:////Injector and ARexx/SCREEN*/AUTO/WAIT/CLOSE, qui vous fournira une fenêtre de console raisonable sur l'écran le plus en avant s'il est public, ou sur l'écran public par défaut.

1.12 2.4 Terminer Injector

Injector peut être terminé de plusieurs façons. Vous pouvez ↩
bien sûr
utiliser le programme Exchange et sélectionner Supprimer Injector. Vous pouvez aussi lancer Injector (le programme principal, pas le programme de

préférences) à nouveau. La dernière façon de quitter Injector est de le faire exécuter la commande

```
CMD_QUIT
.
```

Dans tous les cas, Injector ne peut pas se terminer si un programme ARexx est encore en exécution, car Injector doit attendre le message de réponse d'ARexx (vous ne voulez pas réveiller le gourou n'est-ce pas ? :-). De prochaines versions pourraient implémenter une option "delayed quit" (est-ce vraiment utile?).

1.13 3.1 Lancer le programme de préférences

Pour modifier sa configuration, Injector utilise un programme séparé, appelé le programme de préférences. De cette façon, de la mémoire n'est pas utilisée pour stocker des données inutilisées pendant qu'Injector est actif: le programme de préférences n'est chargé que quand c'est nécessaire.

Le programme de préférences peut être lancé à partir du CLI, du Workbench, ou Injector. Il reconnaît TOUJOURS les tooltypes suivants (même à partir du CLI):

PUBSCREEN=<nom d'écran public>. Ceci définit l'écran sur lequel doit s'ouvrir la fenêtre. Par défaut, c'est *, signifiant que l'écran le plus en avant sera utilisé s'il est public.

CREATEICONS=<YES ou NO>. Ceci instruit le programme de préférences de l'état initial du menu "Créer icônes ?".

ACTION=<USE,SAVE ou EDIT>. Ceci est valide seulement pour les icônes projets dont l'outil par défaut est le programme de préférences, ou donnés par la multi-séléction. De tels projets sont des fichiers de configuration chargés immédiatement par le programme de préférences. Ce tooltype ACTION dit au programme de préférences ce qu'il doit en faire.

USE sauvera le fichier comme la configuration "courante", c'est-à-dire dans le fichier "ENV:Injector.Prefs".

SAVE sauvera le fichier comme la configuration "permanente", c'est-à-dire dans les deux fichiers "ENV:Injector.Prefs" et "ENVARC:Injector.Prefs".

EDIT est la valeur par défaut, il ne sauvegardera rien (encore), mais vous laissera éditer le fichier comme si vous aviez lancé le programme de préférences et séléctionné le menu Ouvrir... .

De plus, il y a quelques options qui peuvent être utilisées sur la ligne de commande, qui SURPASSENT les tooltypes mentionnés ci-dessus. Le format est le suivant:

```
  FILE,PUBSCREEN/K,USE/S,SAVE/S,EDIT/S,NCI=NOCREATEICONS/S
```

FILE est bien sûr le nom du fichier à charger, PUBSCREEN a la même signification que le tooltype du même nom, USE,SAVE et EDIT sont des interrupteurs équivalents aux valeurs correspondantes du tooltype ACTION,

et NOCREATEICONS est équivalent au tooltype CREATEICONS=NO.

1.14 3.2 L'Interface Utilisateur Graphique

Quand le programme de préférences est lancé en mode EDIT, il ↩
ouvre une
fenêtre avec quelques gadgets et menus. Si vous avez l'habitude des programmes de préférences normaux, vous ne devriez pas avoir de problèmes pour l'utiliser. Voici tout de même quelques explications.

L'affichage principal est constitué d'une liste. Elle contient toutes les hotkeys définies. Vous pouvez sélectionner un nom dans cette liste en cliquant dessus, ou en utilisant le raccourci clavier (t pour la version française) avec ou sans SHIFT pour traverser la liste sans utiliser la souris.

Sur la droite, il y a 5 gadgets qui contrôlent l'apparence de la liste. Ils ne changent pas la façon dont Injector traite la liste (excepté peut-être que si deux hotkeys sont en conflit, la première est utilisée). Premier enverra l'élément sélectionné tout en haut de la liste, Dernier l'enverra tout en bas, Haut le déplacera d'une ligne vers le haut, Bas d'une ligne vers le bas, et Trier triera la liste par ordre alphabétique.

Plus bas, il y a quelques gadgets de contrôle. Il y a trois gadgets de chaîne utilisés pour éditer les hotkeys. Pour Injector, une hotkey est une association de trois choses: un nom, une description de touche, et une ligne de commande. Les trois gadgets de chaîne vous permettent d'éditer ces trois champs.

Le champ Nom est utilisé pour afficher la liste et par la commande

EXEC_NAMED
.

Le gadget de chaîne nommé Touche vous permet d'éditer le champ qui dira à la commodities.library quelle touche doit déclencher l'action d'Injector. Ce doit être une chaîne de description valide pour commodities. Voyez

Définir des touches de commande
pour plus d'informations. Après avoir tapé
dans ce gadget de chaîne, le programme de préférences demande à commodities si la description est bonne, et vous alerte dans le cas contraire.

Le gadget de chaîne nommé Commande est là pour contenir la chaîne de commandes qu'Injector doit exécuter quand il reçoit l'évènement correspondant à cette touche. Injector utilise son propre "langage" pour définir de telles actions. Voyez la section
Référence du langage
pour plus
de détails.

Sur la droite des gadgets de chaîne, il y a trois gadgets booléens nommés Créer, Copier et Effacer. Ils sont utilisés respectivement pour créer une nouvelle hotkey (qui apparaît à la fin de la liste), copier une hotkey existante (qui apparaît juste après celle qui est sélectionnée) et effacer

une hotkey existante.

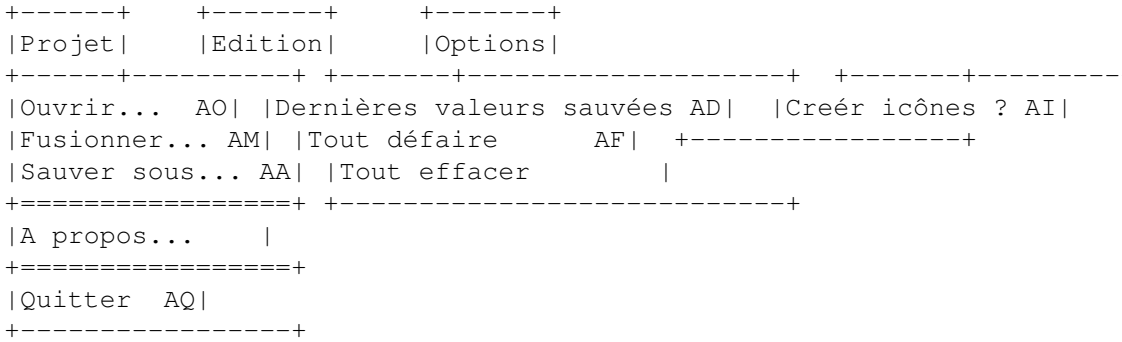
Plus bas encore il y a 5 gadgets. Ils contrôlent le comportement du programme.
Sauver sauvegardera le fichier édité comme configuration permanente. Utiliser le sauvegardera en tant que configuration courante. Tous deux termineront le programme de préférences (à moins qu'une erreur survienne pendant la sauvegarde).

Test changera la configuration courante mais ne terminera pas le programme de préférences, pour que vous puissiez tester si la configuration nouvellement créée vous convient vraiment, et la changer si ce n'est pas le cas.

Aide affichera ce fichier. Il vous faut amigaguide.library V34+ pour cela. A noter que vous devrez changer le nom de ce document si vous voulez que ce soit lui qui soit affiché et pas la version anglaise.

Annuler défera tout ce que vous avez fait. Si vous avez changé la configuration courante (par Tester), alors elle sera remise comme avant que le programme ne soit appelé. Ensuite le programme se termine.

Voilà pour les gadgets. Jetons un coup d'oeil aux menus:



Ouvrir chargera un nouveau fichier de configuration. Celui qui est en cours d'édition sera perdu. Fusionner chargera un nouveau fichier et le fusionnera avec la configuration que vous êtes en train d'éditer. Sauver sous écrira le fichier en cours d'édition dans un nouveau fichier. Pour ces trois actions, vous aurez un file requester d'ASL.

A propos vous informera de la version d'Injector et de mon adresse. Quitter terminera le programme (Attention: aucun "Annuler" ou "Défaire" n'est effectué, une configuration en cours de test restera active).

Dernières valeurs sauvées récupèrera la dernière version sauvegardée du fichier de configuration permanent (la configuration sera chargée à partir de ENVARC:Injector.Prefs). Tout défaire annulera tous les changements que vous avez faits. Tout effacer effacera bien sûr tout le fichier que vous êtes en train d'éditer. Utilisez avec précautions !

Enfin, Créer icônes est un menu booléen qui dit au programme de préférences s'il doit ou non créer une icône pour un fichier qui sera sauvegardé sur disque par la fonction "Sauver sous...". L'icône sera un projet, dont l'image sera celle donnée par ENV:sys/def_prefs.info (ou l'icône projet par défaut si inexistant), l'outil par défaut sera réglé sur le programme de préférences, et ACTION=USE sera en tooltype. De cette façon,

double-cliquer sur cette icône changera la configuration courante d'Injector sans entrer réellement dans le programme de préférences (ce qui ne veut pas dire qu'il n'est pas chargé).

1.15 3.3 Référence du langage

Cette section va vous apprendre comment écrire des commandes pour Injector.
Elles doivent être entrées dans le gadget de chaîne Commande de la fenêtre du programme de préférences quand une hotkey est sélectionnée, ou envoyées par l'intermédiaire d'ARexx au port nommé Injector.

Une commande est constituée d'un mot-clé et d'un argument optionnel. Les commandes qui prennent un argument doivent avoir leur mot-clé suivi immédiatement par l'argument entre parenthèses. Ainsi, il peut être nécessaire pour les programmes ARexx d'entourer les parenthèses par des guillemets, car les premières sont traitées à part par ARexx. Mais les parenthèses doivent rester. Voyez les programmes d'exemple fournis. De multiples commandes sont séparées par des espaces.

Maintenant voici la liste des commandes qu'Injector comprend (celles qui prennent un argument ont () derrière elles):

FREQ_SHOW

TIME_INJDATE()

FREQ_INJFILE

TIME_INJTIME()

FREQ_INJDIR

TIME_INJDAY()

FREQ_INJPATH

EXEC_NAMED()

SREQ_SHOW

EXEC_REXX()

SREQ_INJECT

EXEC_PREFS

CHARS_INJ()

CMD_QUIT

CHARS_CLIP()

CMD_UPDATE

CHARS_ENV()

1.16 La commande FREQ_SHOW

Cette commande montre le file requester. L'utilisateur (vous) peut alors
sélectionner un fichier, qui peut être collé plus tard avec
FREQ_INJFILE
,

FREQ_INJDIR
ou
FREQ_INJPATH
.

L'écran sur lequel le file requester apparaît peut être changé par le
tooltype
PUBSCREEN
.

1.17 La commande FREQ_INJFILE

Cette commande injecte le nom du fichier choisi dans un file requester
précédemment appelé par
FREQ_SHOW
dans la chaîne d'InputEvents. Seule la
dernière composante du nom est prise en compte.

1.18 La commande FREQ_INJDIR

Cette commande est utilisée pour injecter le nom du directory d'un file
requester précedemment appelé. Elle marche comme
FREQ_INJFILE
.

1.19 La commande FREQ_INJPATH

Cette commande est comme un mélange de
FREQ_INJDIR
et
FREQ_INJFILE
mais
prend en charge la gestion du slash ou des deux points entre les deux.
Elle est très utile dans le shell, notemment.

1.20 La commande SREQ_SHOW

Cette commande demande à Injector d'afficher la requête de chaîne. Dans cette requête, vous pouvez taper ce que vous voulez, limité à 80 caractères en longueur.

1.21 La commande SREQ_INJECT

Cette commande injecte le contenu de la requête de chaîne, en supposant qu'elle ait été précédemment appelée par SREQ_SHOW. C'est utile si vous devez taper beaucoup de fois la même chose.

1.22 La commande CHARS_INJ

Cette commande injecte la chaîne de caractères constante, qui est donnée en argument, dans la chaîne des InputEvents. Quelques caractères spéciaux sont reconnus pour cette commande (ce sont les mêmes que ceux utilisés par CatComp):

```
  \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  Tabulation horizontale (ASCII 9)
  \v  Tabulation verticale (ASCII 11)
  \xNN  Caractère dont le code ASCII est NN en hexadécimal.
  \NNN  Caractère dont le code ASCII est NNN en octal.
  \ Backslash
  \)  Parenthèse de fermeture (pas la fin des arguments).
```

Tous sont en minuscules.

1.23 La commande CHARS_CLIP

Cette commande injecte le contenu du calepin (clipboard) système. Pour être précis, tous les chunks CHRS trouvés auront leur contenu injecté dans la chaîne d'InputEvents, permettant ainsi une façon simple de partager des données entre les applications (vous pouvez couper du texte dans le shell et l'injecter dans un traitement de texte qui ne gère pas le calepin.

L'argument est l'unité du calepin, de 0 à 255. C'est en général 0 (pour le shell etc...), pas pas nécessairement.

1.24 La commande CHARS_ENV

Cette commande injecte le contenu d'une variable d'environnement dont le nom est donné en argument. Injector ne fait pas de vérifications, alors évitez les fichiers binaires !

1.25 La commande TIME_INJDATE

Cette commande injecte la date, selon le format demandé par l'argument.

Si la locale.library n'a pu être ouverte avec succès, l'argument est forcé à 2. Sinon, voici les valeurs possibles (les autres valeurs sont réservée pour une extension future):

0: format court, selon la locale courante
(ex: 21/04/94)

1: format long, selon la locale courante
(ex: Jeudi 21 Avril 1994)

2: format DOS. C'est comme cela que le DOS affiche la date.
(ex:21-Avr-94)

1.26 La commande TIME_INJTIME

Ceci vous permet d'injecter la date courante. L'argument est un numéro de
format similaire à celui pris par
TIME_INJDATE
.

ex: 0: 12h06
1: 12h06
2: 12:06:21
(la locale française ne fait pas la différence entre les formats long et court pour l'heure).

1.27 La commande TIME_INJDAY

Voici la dernière commande TIME_: elle colle le nom du jour. L'argument
est toujours un numéro de format (voyez
TIME_INJDATE
).

ex 0: Jeu
1: Jeudi
2: Jeudi

1.28 La commande EXEC_NAMED

Cette commande execute une hotkey comme si la touche correspondante avait été pressée par l'utilisateur. De cette façon vous pouvez faire des interprétations à la GOSUB des hotkeys. L'argument est bien évidemment le nom de la hotkey à exécuter. Il y a une sécurité dans Injector: quand une hotkey est exécutée, un bit spécial est mis à 1 dans sa structure, ce qui empêche les appels récursifs (vous aurez juste un message).

1.29 La commande EXEC_REXX

Cette commande demande à Injector de lancer le programme ARexx dont le nom est en argument. Le programme aura son adresse hôte par défaut égale à "Injector". L'extension par défaut pour de tels programmes est .ijctr .

Grâce à cette commande (et à ARexx !), vous pouvez générer des macros complexes, avec des tests, etc etc etc...

1.30 La commande EXEC_PREFS

Cette commande demande à Injector de lancer le programme de ↩
préférences.
Ceci est réalisé par la fonction SystemTagList() de la dos.library, donc le programme de préférences doit se trouver dans votre chemin. Le nom de fichier par défaut est "InjectorPrefs", mais il peut être modifié par le tooltype

PREFSPATH

.

1.31 La commande CMD_QUIT

L'utilisation de cette commande est évidente: quand il la reçoit, et pourvu que ce soit possible, Injector se suicide. Ce n'est pas possible par exemple quand des programmes ARexx tournent encore.

1.32 La commande CMD_UPDATE

Cette commande n'est pas utile dans la plupart des cas. Quand il la reçoit, Injector recharge sa configuration. C'est inutile dans la plupart des cas car, comme le programme IPrefs, Injector utilise les possibilités de notification d'AmigaDOS pour détecter automatiquement les altérations du fichier de préférences (par le programme de préférences par exemple). Généralement, ENV: est assigné au Ram disk, donc ce n'est pas un problème. Mais tous les gestionnaires ne gèrent pas la notification, alors, si quelqu'un utilise un assign sur un système de fichier en réseau par exemple, il doit utiliser cette commande pour qu'Injector mette à jour sa configuration.

Quand cette commande est reçue, Injector ne continue pas l'exécution de la ligne courante, même s'il restait des commandes.

1.33 4.1 Définir des touches de commande

Le texte qui suit ne dépend que du comportement de la commodities.library et pas de celui d'Injector. C'est comme cela que commodities comprendra la combinaison de touches à laquelle vous voulez associer une action.
Une chaîne de description est constituée comme suit:

[<classe>] {[-][<qualificateurs>]} [-][upstroke] [<code>]

Tous les mots-clés sont insensibles aux majuscules.

classe: doit être une classe d'InputEvent. Les classes supportées sont rawkey pour les évènements clavier, rawmouse pour les évènements souris, diskinserted et diskremoved. Par défaut c'est rawkey, qui doit généralement être utilisée.

qualificateurs: c'est une série de mots-clés représentant l'état des qualificateurs du clavier (Shift,Alt etc...). Voici une liste de mots-clés connus. Ceux qui sont marqués par un * sont nouveaux pour la version 38 de commodities.library.

lshift,left_shift *: Touche shift gauche.
rshift,right_shift *: Touche shift droite.
shift: Une touche shift.
capslock,caps_lock *: Touche Caps Lock.
caps: Touche Caps Lock ou shift.
control,ctrl *: Touche Control.
lalt,left_alt *: Touche Alt gauche.
ralt,right_alt *: Touche Alt droite.
alt: Une touche Alt.
lcommand,lamiga *,left_amiga *,left_command *: Touche amiga gauche.
rcommand,ramiga *,right_amiga *,right_command *:Touche amiga droite.
numericpad,numpad *,num_pad *,numeric_pad *: Pour les touches du clavier numérique.
leftbutton,lbutton *,left_button *: Bouton souris gauche.(1)
midbutton,mbutton *,middlebutton *,middle_button *: Bouton souris milieu.(1)
rbutton:,rightbutton *,right_button *: Bouton souris droit.(1)
repeat: Répétition active.(2)

Notes: (1) la commodities.library V37 contenait une erreur qui l'empêchait d'utiliser leftbutton,midbutton et rbutton comme qualificateirs. Ce fut réparé pour la version 38.
(2) pour la classe rawkey uniquement.
(3) si une touche de commande doit être insensible à l'état d'un qualificateur, placez un - avant son nom.

upstroke: Normalement un évènement est généré seulement quand la touche est pressée. Vous pouvez changer ce comportement en ajoutant upstroke. Cela générera un évènement uniquement quand la touche sera relachée. Si les faits de presser et de relacher doivent tous deux générer un évènement, utilisez -upstroke.

```
code:   Ceux-ci  sont  significatifs  uniquement pour les classes rawkey et
rawmouse.   Pour rawkey,voici les codes de touches, * sont nouveaux pour la
version 38 de commodities.
a to z, 0 to 9:       Caractères ASCII normaux.
f1 to f10,f11 and f12:      Touches de fonction.
up,cursor_up *:       Flèche haut.
down,cursor_down *:     Flèche bas.
left,cursor_left *:     Flèche gauche.
right,cursor_right *:     Flèche droite.
esc,escape *:       Touche Esc.
backspace       Espace arrière.
del         Touche Del.
help          Touche Help.
tab         Touche Tab.
comma         Touche virgule (,).
return          Touche retour chariot.
space,spacebar *      Barre d'espace.
enter         Touche Enter.(4)
insert *        Touche pad 0.(4)
delete *        Touche pad 1.(4)
page_up *       Touche pad 9.(4)
page_down *       Touche pad 3.(4)
home *          Touche pad 7.(4)
end *         Touche pad 1.(4)

Notes:  (4) à utiliser avec le qualificateur numericpad.

Pour rawmouse, les codes valides sont:  (uniquement pour Commodities V38):
mouse_leftpress:      Bouton gauche pressé.(5)
mouse_middlepress:      Bouton du milieu pressé.(5)
mouse_rightpress:     Bouton droit pressé.(5)

Notes:  (5) vous devez utiliser aussi le qualificateur correspondant.
```

1.34 4.2 Historique

```
Revision V2.00
--------------
    Roulement de tambours... Ca y est enfin (plus de 6 mois après
    la première distribution!). J'ai totalement ré-écrit le programme,
    avec plein de nouvelles options puissantes, un joli programme
    de préférences, support ARexx complet, localisation... Il a
    peu de choses en commun avec les versions 1.x, alors lisez la
    doc !

Revision V1.16
--------------
    Commodore a vraiment foiré avec sa mauvaise documentation de la
    fonction CreateNewProc(): la valeur par défaut de NP_FreeSegList
    est FAUX! Cela causait d'importantes pertes de mémoire dans les
    versions précédentes d'Injector...

Revision V1.15
--------------
    Réparé un bug vicieux: Echec de la libération d'un écran public
```

dans certains cas.

Revision V1.14

Réparé un bug stupide: PASTEFILEKEY et PASTEDIRKEY étaient inversés.

Revision V1.13

Ca y est enfin ! Je suis arrivé à la toute première distribution publique, en SHAREWARE. Comme d'habitude, j'ai fait un peu de rangement et tout rassemblé dans un beau directory avec de jolis fichiers readme. (10-Oct-93 14:31:10)

Revision V1.12

Ajouté une chaîne de version 2.0 et écrit des doc AmigaGuide Pas de doc ASCII pour ce programme, il vous faut AmigaGuide (Fish 870) pour les lire. Optimisé le placement de la fenêtre icône (droite du requester). La distribution publique est pour la prochaine révision.

Revision V1.11

Le tooltype CXPRI foirait,réparé maintenant. J'ai écrit un catalogue français et j'ai réparé l'interface avec locale. Fait quelques réparations et optimisations diverses. Je prévois une distribution publique bientôt. Promis!

Revision V1.10

Eh bien, j'ai enlevé la traduction automatiquement (implémentée avec les objects translates de Commodities).Avantages: cela raccourci le programme, commodities ne cause plus de gurus étranges en accédant de façon asynchrone des données au mauvais moment. Inconvénient: Impossible d'utiliser les hotkeys quand le requester est actif.

Revision V1.09

J'ai fait marcher les tooltypes PubScreen et IconPath et j'ai enlevé quelques bugs. Je pense à écrire un fichier de doc maintenant...

Revision V1.08

J'ai fait un squelette pour le support des options CLI et tooltypes. Options supportées pour l'instant: toutes les hotkeys,priorité Commodities. Bientôt: IconPath and PubScreen.

Revision V1.07

Ecrit mes propres routines InvertString()/FreeIEvents() pour qu'Injector n'interprète plus mal les noms de fichier avec des <>. Aussi réparé l'auto-détachant du Workbench pour permettre l'utilisation du tiroir WBStartup.

```
Revision V1.06
--------------
    Maintenant injector se détache du CLI, plus besoin d'utiliser
    Run. Si la fenêtre de l'icône est déplacée, la prochaine fois
    elle s'ouvre au même endroit.

Revision V1.05
--------------
    L'icône affiché est finalement un gadget réellement sélec-
    tionnable, permettant de voir les deux états actif et inactif.
    Notez qu'il peut y avoir une différence avec les icônes du
    Workbench, qui a un mode "floodfill" non supporté par les
    gadgets Intuition standards. Peut-être plus tard.

Revision V1.04
--------------
    Ah, une cool nouvelle option: le requester d'icône peut
    maintenant afficher les icônes eux-mêmes dans leur propre
    petite fenêtre, en utilisant encore un joli ExtraButton.
    Pour l'instant, l'icône n'est pas un vrai gadget (à peine
    une image), mais ce sera fait bientôt.

Revision V1.03
--------------
    Ca y est, Injector est maintenat capable de copier des icônes
    pour des tiroirs,choisissez simplement un nom de fichier vide.

Revision V1.02
--------------
    Ajouté une nouvelle option intéressante: Injector peut main-
    tenant copier des icônes d'un endroit à un autre, en utilisant
    l'option ExtraButton de la kd_freq. C'est très utile. Il faut
    encore que je m'occupe des icônes de tiroirs.

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

1.35 4.3 Contacter l'auteur

Je vous rappelle qu'Injector est SHAREWARE. Je sais que la plupart d'entre vous ne me paieront rien après leur période d'évaluation, mais sachez que j'ai passé beaucoup de temps à écrire et débugger ce programme, et que je voudrais juste une petite compensation qui me permettra d'acheter une bonne config hardware. Payer $10 vous fournira un support pour les prochaines versions. Je garde le source au chaud pour ceux d'entre vous qui sont intéressés par la programmation du merveilleux système d'exploitation de l'amiga, pour $10 de plus.

Pour tout ce qui concerne l'enregistrement, les commentaires, rapport de bugs, demandes d'améliorations, cartes postales, vous pouvez m'écrire à:

Frédéric DELACROIX
5 rue d'Artres
59269 QUERENAING

FRANCE.