

**Injector.doc**

**COLLABORATORS**

	<i>TITLE :</i> Injector.doc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 7, 2024	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Injector.doc</b>	<b>1</b>
1.1	Documentation d'Injector - Français	1
1.2	1.1 Distribution et décharge	1
1.3	1.2 Crédits et remerciements	3
1.4	1.3 Configuration minimale	3
1.5	2.1 Introduction	4
1.6	2.2 Lancer Injector	4
1.7	2.3 Les tooltypes	5
1.8	Le tooltype PUBSCREEN	5
1.9	Le tooltype CX_PRIORITY	5
1.10	Le tooltype PREFSPATH	5
1.11	Le tooltype AREXXCONSOLE	5
1.12	Le tooltype BREAKKEY	5
1.13	Le tooltype TOOLPRI	6
1.14	2.4 Terminer Injector	6
1.15	3.1 Lancer le programme de préférences	6
1.16	3.2 L'Interface Utilisateur Graphique	7
1.17	3.3 Référence du langage	10
1.18	La commande <code>FREQ_SHOW</code>	10
1.19	La commande <code>FREQ_INJFILE</code>	10
1.20	La commande <code>FREQ_INJDIR</code>	11
1.21	La commande <code>FREQ_INJPATH</code>	11
1.22	La commande <code>SREQ_SHOW</code>	11
1.23	La commande <code>SREQ_INJECT</code>	11
1.24	La commande <code>CHARS_INJ()</code>	11
1.25	La commande <code>CHARS_CLIP()</code>	12
1.26	La commande <code>CHARS_ENV()</code>	12
1.27	La commande <code>TIME_INJDATE()</code>	12
1.28	La commande <code>TIME_INJTIME()</code>	12
1.29	La commande <code>TIME_INJDAY()</code>	12

---

---

1.30	La commande TIME_LOCALIZED()	13
1.31	La commande EXEC_NAMED()	14
1.32	La commande EXEC_REXX()	14
1.33	La commande EXEC_PREFS	14
1.34	La commande CMD_QUIT	14
1.35	La commande CMD_UPDATE	15
1.36	La commande CMD_NEWPREFS()	15
1.37	La commande KREQ_SHOW	15
1.38	La commande KREQ_INJECT	16
1.39	La commande PUB_SHOWLIST	16
1.40	La commande PUB_INJECT	16
1.41	La command PUB_FRONT	17
1.42	La commande CHARS_RAWKEY()	17
1.43	La commande CHARS_FILE()	17
1.44	3.4 Versions des préférences	17
1.45	4.1 Définir des touches de commande	18
1.46	4.2 Historique	19
1.47	4.3 Contacter l'auteur	25

---

# Chapter 1

## Injector.doc

### 1.1 Documentation d'Injector - Français

Bienvenue à Injector 2.32 copyright 1993-94 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 Configuration minimale
- 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
  - 3.4 Versions des préférences
- 4 APPENDICE
  - 4.1 Définir des touches de commandes
  - 4.2 Historique
  - 4.3 Contacter l'auteur <- Faites-le !

### 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'apprécierais 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.

Je ne distribuerai pas le fichier cd avec ce programme. Si vous voulez une traduction, envoyez-moi le fichier ct avec votre contribution...

Injector a maintenant reçu des icônes dignes de ce nom. Celles-ci ont été dessinées par Christophe Peugnet (de ProMedia) qui en possède le copyright, elles ne sont distribuables qu'avec Injector. Dans le tiroir OtherIcons, il y a deux icônes supplémentaires, dessinées et copyrightées par Laurent Pazdziej, distribuables seulement avec Injector.

J'ai oublié de mentionner dans les versions précédentes (je m'en excuse...) que les icônes que j'ai utilisées pour les tiroirs et d'autres fichiers sont tirées d'IconPack, qui est une collection de jolies icônes dessinées

et copyrightées par Tom Ekström.

## 1.3 1.2 Crédits et remerciements

Injector a été écrit avec le merveilleux Devpac 3 d'Hisoft sur un vieil 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.

Remerciements à: Nico François pour reqtools.library  
Khalid Aldoseri pour sa kd\_freq.library  
William S. Hawes pour ARexx  
Hisoft pour Devpac 3  
Commodore pour l'amiga et son OS  
CEI pour le sauvetage de l'amiga (?)  
J.P. Deloffre et C.Peugnet de l'association ProMedia  
Christophe Peugnet pour ses belles icônes  
Laurent Pazdziej pour d'autres icônes  
Yves Quiquempois pour les tests sous Enforcer et idées  
d'amiéliorations  
AmigaNews.

## 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 V4.0+ 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+. La V39 est encore mieux car exempte de bugs.

- 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 ont 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 fenêtre 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 (n'oubliez pas de vous enregistrer). 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, AREXXCONSOLE, BREAKKEY et TOOLPRI.

## 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 raisonnable sur l'écran le plus en avant s'il est public, ou sur l'écran public par défaut.

## 1.12 Le tooltype BREAKKEY

Ce tooltip est nouveau pour la version 2.28. C'est une chaîne de description pour commodities utilisée par Injector pour savoir quelle hotkey doit arrêter l'action d'Injector.

Cette possibilité de "touche d'arrêt" a été ajoutée pour empêcher Injector d'injecter toutes sortes de choses et de devenir incontrôlable. Si, par exemple, vous avez choisi le mauvais fichier, vous pouvez toujours arrêter l'injection avec cette touche.

Contrairement aux hotkeys "normales", l'évènement n'est PAS retiré de la chaîne. De cette façon, vous pouvez quitter Injector en même temps, ou arrêter un autre programme comme WindowKey en même temps. Ceci peut avoir des effets désastreux si l'application qui possède la fenêtre active reconnaît ce message.

La valeur par défaut est: lcommand rcommand lshift rshift q

## 1.13 Le tooltip TOOLPRI

Ce tooltip n'est reconnu que pour la version 2.30. C'est un tooltip standard, tel qu'il est compris par le Workbench, pour régler la priorité de la tâche d'Injector. Cependant, comme Injector lance son propre processus, la priorité était toujours de -1 dans les versions précédentes. Maintenant vous pouvez la donner avec ce tooltip, qui est aussi reconnu depuis le CLI.

## 1.14 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?).

## 1.15 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 tooltips 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-sélection. De tels projets sont des fichiers de configuration chargés immédiatement par le programme de préférences. Ce tootype 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 sélectionné le menu Ouvrir... .

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

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

FILE est bien sûr le nom du fichier à charger, PUBSCREEN a la même signification que le tootype du même nom, USE,SAVE et EDIT sont des interrupteurs équivalents aux valeurs correspondantes du tootype ACTION, et NOCREATEICONS est équivalent au tootype CREATEICONS=NO. CREATEICONS a été introduit à partir de la version 2.10 et est équivalent au tootype CREATEICONS=YES. Les deux dernières options peuvent être utilisées pour surpasser les valeurs par défaut, je suppose qu'elles ne le seront pas beaucoup.

## 1.16 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. A partir de la version 2.11, vous pouvez aussi utiliser les flèches vers le haut et vers le bas pour vous déplacer d'une entrée dans la direction correspondante, d'une page avec shift, ou pour aller au début ou à la fin de la liste avec control.

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.

Nouveau à partir de la version 2.10, il y a une checkbox sur la droite du gadget descripteur de touche. Si elle est sélectionnée, alors la hotkey est autorisée (comme c'était le cas avant la version 2.10). Si elle n'est pas sélectionnée, alors la hotkey n'est pas disponible. Elle l'est toutefois toujours grâce à la commande EXEC\_NAMED().

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. A partir de la version 2.10, si la description est correcte, alors la checkbox sera automatiquement sélectionnée, et désélectionnée sinon.

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:

```
+-----+ +-----+ +-----+
|Projet| |Edition| |Options|
+-----+-----+ +-----+-----+ +-----+-----+
|Ouvrir... AO| |Dernières valeurs sauveés AD| |Créer icônes ? AI|
|Fusionner... AM| |Tout défaire AF| +-----+
|Sauver sous... AA| |Tout effacer |
+=====+ +-----+
|A propos... |
+=====+
|Quitter AQ|
+-----+
```

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 sauveés 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 tootype. 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).

A partir de la version 2.29, lancer une nouvelle fois le programme de préférences aura pour effet d'activer, amener en avant et "dé-zoomer" la fenêtre si nécessaire. Lui envoyer un CTRL-F aura le même effet. Envoyer un CTRL-C le terminera.

A partir de la version 2.32, le programme de préférences s'adapte à la fonte. La checkbox peut paraître bizarre avec une autre fonte que topaz/8 sous OS2.0 car gadtools ne la recalibre pas à la taille du gadget automatiquement. Cela a été réparé pour l'OS3.0.

## 1.17 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  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 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.

## 1.19 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.

## 1.20 La commande `FREQ_INJDIR`

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

## 1.21 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, notamment.

## 1.22 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.

## 1.23 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. A partir de la V2.20, vous pouvez utiliser des caractères spéciaux dans le requester, comme les newlines (`\n`), bells (`\a`) etc... La liste de ces caractères spéciaux est donnée dans la commande `CHARS_INJ()`.

## 1.24 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.

---

## 1.25 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.

## 1.26 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 !

## 1.27 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 réservées 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)

## 1.28 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).

## 1.29 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
```

### 1.30 La commande TIME\_LOCALIZED()

Cette commande est nouvelle depuis la version 2.03 d'Injector. Pour qu'elle marche, il vous faut la locale.library installée dans votre système. Aucune injection ne sera faite si vous ne l'avez pas.

Cette commande est similaire aux autres commandes TIME\_; elle vous permet d'injecter l'heure avec un format personnel, selon la fonction FormatDate() de la locale.library. L'argument est une chaîne de caractères qui contient éventuellement des codes de formatage qui seront remplacés selon les règles suivantes. Tous ces codes commencent avec un signe %, utilisez %% pour un vrai %. Ceux d'entre vous qui connaissent printf() du C (ou RawDoFmt() d'exec), ne seront pas dépaysés. Cependant, les codes sont très différents:

```
%a - nom abrégé du jour de la semaine
%A - nom du jour de la semaine
%b - nom abrégé du mois
%B - nom du mois
%c - identique à "%a %b %d %H:%M%S %Y"
%C - identique à "%a %b %e %T %Z %Y"
%d - numéro du jour précédé par des 0
%D - identique à "%m/%d/%y"
%e - numéro du jour précédé par des espaces
%h - nom du mois abrégé
%H - heure en utilisant le mode 24 heures, précédée par 0
%I - heure en utilisant le mode 12 heures, précédée par 0
%j - date julienne
%m - numéro du mois précédé par des 0
%M - nombre de minutes précédé par des 0
%n - Passe à la ligne
%p - AM ou PM
%q - heure en utilisant le mode 24 heures
%Q - heure en utilisant le mode 12 heures
%r - identique à "%I:%M:%S %p"
%R - identique à "%H:%M"
%S - nombre de secondes précédé par des 0
%t - tabulation
%T - identique à "%H:%M:%S"
%U - numéro de la semaine, avec dimanche comme premier jour
%w - numéro du jour de la semaine
%W - numéro de la semaine, avec lundi comme premier jour
%x - identique à "%m/%d/%y"
%X - identique à "%H:%M:%S"
%y - année à deux chiffres
%Y - année à quatre chiffres
```

Donc, par exemple, la commande

```
TIME_LOCALIZED(Aujourd'hui %A %e %B %Y, il est %H heures.)
```

injectera ceci (du moins au moment où je tape :-)

Aujourd'hui Mardi 21 Juin 1994, il est 19 heures.

### 1.31 La commande EXEC\_NAMED()

Cette commande exécute 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).

### 1.32 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...

### 1.33 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.

A partir de la version 2.11, ceci peut aussi être fait en sélectionnant Montrer dans le programme commodities Exchange. A partir de la version 2.29, sélectionner Cacher terminera le programme de préférences (en lui envoyant un CTRL-C).

### 1.34 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.

### 1.35 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.

### 1.36 La commande CMD\_NEWPREFS()

Cette commande changera de fichier de configuration. L'argument doit être un chemin valide jusqu'au fichier en question. Vous pouvez créer un tel fichier en sélectionnant le menu Sauver sous... dans le programme de configuration.

Injector fait le changement en invoquant le programme de préférences avec l'argument et le mot-clé USE, donc le programme de préférences doit être accessible. Vous pouvez utiliser l'option PREFSPATH pour cela.

Cette commande existe depuis la version 2.02 d'Injector, utilisable depuis la version 2.03.

### 1.37 La commande KREQ\_SHOW

Cette commande ouvre la fenêtre du requester de touche. Cette fenêtre s'ouvrira sur l'écran spécifié par le tootype PUBSCREEN, sa largeur sera le huitième de la largeur totale de l'écran, ainsi que sa hauteur, et elle sera positionnée juste sous le pointeur de la souris. Cette fenêtre attend que vous produisiez un évènement, comme insérer une disquette, retirer une disquette, presser une touche ou un bouton de la souris (qualifié), rendre la fenêtre inactive.

Pour pouvoir recevoir des évènements vraiment qualifiés, les touches qualifieurs (shift, alt, control, etc) sont filtrées (elles ne produisent pas d'évènement rawkey), ainsi que les boutons de la souris lorsqu'ils sont pressés (ils produisent un évènement rawmouse quand ils sont relâchés). Apparemment, Intuition ne semble pas envoyer de masque de qualifieurs pour les classes IDCMP\_DISKINSERTED et IDCMP\_DISKREMOVED donc les qualifieurs sont ignorés quand de tels évènements interviennent (pas ma faute). De plus, comme l'entrée se fait à travers Intuition, toutes les touches utilisées par les InputHandlers de priorité supérieure à celle d'Intuition (comme les brokers de commodities) ne seront pas passés à la fenêtre, de tels évènements ne peuvent pas être repérés.

Inactiver la fenêtre est considéré par Injector comme une demande d'annulation. Ceci peut être fait en cliquant avec le bouton gauche de la souris en dehors de la fenêtre. C'est pourquoi quand vous voulez produire des évènements concernant le bouton gauche de la souris (en tant que qualifieur ou évènement rawmouse), vous devez cliquer à l'intérieur de la fenêtre. A partir de la version 2.15, il y a aussi un temps limite d'environ 10s.

Après avoir satisfait la fenêtre, vous pouvez transformer cet évènement en une chaîne de description pour commodities (voyez l'appendice 4.1) avec la commande KREQ\_INJECT. Cette commande est nouvelle pour la version 2.12 d'Injector.

### 1.38 La commande KREQ\_INJECT

Cette commande est utilisée en conjonction avec KREQ\_SHOW: elle injecte la chaîne de description pour commodities pour l'évènement que vous lui avez donné. Ceci est utile pour définir des hotkeys pour toolmanager, un blanker, et, bien sûr, le programme de préférences d'Injector.

Quelques options ne sont pas disponibles (comme le qualifieur repeat, bien sûr), mais vous pouvez toujours éditer la chaîne produite...

Cette commande est nouvelle pour la version 2.12 d'Injector.

### 1.39 La commande PUB\_SHOWLIST

Cette commande, nouvelle pour la version 2.17, est utilisée pour afficher une petite fenêtre avec une liste de tous les écrans publics actuellement ouverts. Vous pouvez en sélectionner un en cliquant sur un nom dans la liste et sur Choisir, ou en double-cliquant un nom. La liste peut aussi être contrôlée au clavier (flèches, RETURN pour choisir, ESC pour annuler). Les gadgets nommés Choisir et Annuler sont assez explicites. Le gadget nommé Frontal est utilisé pour sélectionner automatiquement dans la liste le nom de l'écran frontal s'il est public.

La fenêtre se rappellera de sa position après sa fermeture, et sélectionnera automatiquement le nom de la liste identique à celui contenu dans le buffer (entré par PUB\_SHOWLIST ou, pour la version 2.18, par PUB\_FRONT).

Après avoir sélectionné un nom dans la liste, vous pouvez l'injecter avec la commande PUB\_INJECT.

A partir de la version 2.32, cette requête s'adapte à la fonte.

### 1.40 La commande PUB\_INJECT

Cette commande va injecter un nom d'écran public précédemment choisi avec PUB\_SHOWLIST ou PUB\_FRONT. Cette est nouvelle pour la V2.17.

---

Par exemple, cette hotkey pourrait être utile dans un shell:

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

(voyez le fichier de configuration fourni).

## 1.41 La command PUB\_FRONT

Cette commande est nouvelle pour la V2.18. Son travail est de remplir le tampon de nom d'écran public avec le nom de l'écran public le plus en avant. Si l'écran le plus en avant n'est pas public, le tampon reste inchangé. Cette commande agit comme si vous aviez utilisé PUB\_SHOWLIST et sélectionné "Frontal" et "Choisir". Mais la liste n'est pas affichée. Vous pouvez injecter le résultat avec PUB\_INJECT.

## 1.42 La commande CHARS\_RAWKEY()

Cette commande est nouvelle pour la version 2.23. L'argument donné est une chaîne de description pour commodities (comme décrit dans l'appendice 4.1). Elle est utilisée pour décrire la touche à injecter, que ce soit un caractère affichable ou non affichable, ou même pas de caractère du tout, c'est-à-dire que les chaînes comme "rawkey lcommand rcommand control lshift rshift numpad rbutton lbutton mbutton \*" sont parfaitement valides (bien que pas vraiment utilisables :-). Vous pouvez même utiliser des événements rawmouse.

Comme les événements sont injectés avec la commande IND\_WRITEEVENT de l'input.device, de telles touches sont propagées à tous les handlers, y compris, et c'est là où CHARS\_RAWKEY() devient utile, aux brokers de commodities. Vous pouvez ainsi définir des synonymes pour les hotkeys ou cumuler beaucoup d'actions sur une hotkey. Personnellement, j'utilise la hotkey nommée "Toolmanager docs" dans le fichier de configuration fourni en exemple pour ouvrir tous mes docks toolmanager en même temps.

## 1.43 La commande CHARS\_FILE()

Cette commande est nouvelle pour la version 2.27. Elle permet d'injecter le contenu du fichier dont le nom est donné en argument. Attention, aucune vérification n'est faite, évitez les fichiers binaires !

Un nom spécial est reconnu: \* signifie le fichier qui a été choisi dans une requête de fichier précédemment appelée grâce à FREQ\_SHOW.

## 1.44 3.4 Versions des préférences

---

Pour la V2.10, j'ai dû incorporer de nouvelles options dans le fichier de préférences, ce qui signifie 2 octets supplémentaires de données par hotkey. C'est pourquoi le numéro de version dans le fichier de préférences a augmenté, et c'est pourquoi Injector ne peut pas lire les fichiers plus vieux. Cependant, le programme de préférences peut lire les vieux fichiers, vous pouvez l'utiliser pour convertir vos fichiers de préférences pré-V2.10 en nouveaux en les chargeant dans le programme de préférences et en les resauvegardant.

## 1.45 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. A partir de la version 2.12, les commandes `KREQ_SHOW/KREQ_INJECT` pourraient aussi se révéler utiles...

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 qualificateurs. 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.

## 1.46 4.2 Historique

Revision V2.32

-----

Enfin sensible à la taille des fontes! Utilisera topaz/8 seulement si la fonte de l'écran est proportionnelle. Vrai pour le programme de préférences et le requester d'écrans publics.

Revision V2.31

-----

Adapté à la version 3.02 d'errmsg.library, causant un raccourcissement des exécutables, des messages enlevés. Je vais distribuer seulement le fichier bibliothèque lui-même et le script d'installation pour réduire la taille de l'archive.

Revision V2.30

-----  
Reconnaît maintenant le tooltype TOOLPRI (comme le Workbench).

Revision V2.29

-----  
Dans le programme de préférences, tab active le gadget nom. Le programme de préférences quittera à la réception de CTRL-C, et le message de commodities Cacher est maintenant reconnu.

Revision V2.28

-----  
Implémenté la "touche d'arrêt".

Revision V2.27

-----  
Implémenté CHARS\_FILE.

Revision V2.26

-----  
Bug réparé: l'injection du key requester se plantait quand l'utilisateur utilisait la barre d'espace. (Bug rapporté par Y.Quiquempois).

Revision V2.25

-----  
Ajouté un gadget de zoom pour le programme de préférences.

Revision V2.24

-----  
Réparé un bug mineur qui causaient l'injection d'un caractère nul par les commandes TIME\_ basées sur la locale à la fin d'une chaîne. Réparé un problème avec la requête d'écrans publics.

Revision V2.23

-----  
Implémenté CHARS\_RAWKEY.

Revision V2.22

-----  
Injector a encore des problèmes pour injecter des chaînes longues sur des machines rapides, c'est pourquoi j'utilise maintenant input.device/IND\_WRITEEVENT au lieu de commodities.library/AddIEvents(). J'espère que ça marche maintenant !

Revision V2.21

-----  
Utilise maintenant les crayons de DrawInfo pour le motif de remplissage au lieu de systématiquement utiliser les crayons 0 et 2.

Revision V2.20

-----

---

Vous pouvez maintenant utiliser les caractères avec backslash (\n, \a etc) dans la requête de chaîne. Quelques optimisations diverses.

Revision V2.19

-----  
Réécrit la fonction d'injection de caractères: elle utilise maintenant MapANSI() directement, au lieu d'InvertKeyMap(). J'espère que cela règle un bug mangeur de mots rapporté par un de mes amis.

Revision V2.18

-----  
Implémenté PUB\_FRONT.

Revision V2.17

-----  
Implémenté la requête d'écran public (commandes PUB\_SHOWLIST et PUB\_INJECT).

Revision V2.16

-----  
Ajouté un joli motif de remplissage pour la requête de touche.

Revision V2.15

-----  
Ajouté de la couleur pour le fon des gadgets de chaîne sélectionnés dans le programme de prefs) et un temps limite pour le requester de touche.

Revision V2.14

-----  
Changé tous les `move.l 4,w,a6` en `move.l Exec.Base(pc),a6 ...`  
Je sais que j'aurais dû le faire depuis le début, enfin ça y est.

Revision V2.13

-----  
Optimisé la fenêtre du requester de touche: taille=écran/8, et position sous le pointeur. J'ai réglé le cas des caractères non affichables: les qualifieurs autres que shift sont éliminés avant MapRawKey(), ceci devrait être parfait pour toutes les keymaps actuelles.

Revision V2.12

-----  
Implémenté les puissantes fonctions KREQ\_SHOW et KREQ\_INJECT.  
Il faut encore que je règle le cas des caractères non affichables (comme ctrl-w).

Revision V2.11

-----  
Injector supporte le message SHOW de commodities, il appelle le programme de préférences. Les flèches haut/bas peuvent maintenant être utilisées pour scroller dans la liste.

Revision V2.10

-----  
Les hotkeys peuvent maintenant être désactivées grâce à une

---

nouvelle Checkbox dans l'IUG du programme de préférences. Cela a causé un changement dans la structure du fichier de préférences, mais le programme de préférences peut toujours lire les anciens fichiers. Les touches désactivées sont toujours disponibles par EXEC\_NAMED. Implémenté l'option CLI CREATEICONS.

Revision V2.09

-----  
Réparé un hit Mungwall lors de la sélection de multiples menus.

Revision V2.08

-----  
Changé le directory par défaut de Ouvrir, Sauver sous et Fusionner depuis presets/ vers PROGDIR:presets/ . Ceci est beaucoup plus sensé puisque le programme de préférences doit être stocké dans le directory prefs.

Revision V2.07

-----  
Ce bug stupide est vraiment bizarre. J'ai résolu ce problème dingue en ouvrant amigaguide.library au démarrage au lieu de de l'aide. J'ai aussi réparé le report d'erreur d'AmigaGuide (OpenAmigaGuideA() retourne un AMIGAGUIDECONTEXT même si une erreur est survenue)...

Revision V2.06

-----  
Bug réparé: quand le fichier d'aide n'était pas disponible et que l'utilisateur cliquait deux fois (!) sur le bouton Aide, le programme de prefs se plantait. Merci à Yves Quiquempois d'avoir rapporté ce bug bizarre.

Revision V2.05

-----  
Réparé un léger bug: si la reqtools.library ne pouvait être ouverte, l'iffparse.library n'était pas fermée.

Revision V2.04

-----  
Réduit la taille de l'exécutable en enlevant quelques Reloc32 et en optimisant un peu le code.

Revision V2.03

-----  
Implémenté la nouvelle fonction TIME\_LOCALIZED, débogué CMD\_NEWPREFS.

Revision V2.02

-----  
Implémenté la fonction CMD\_NEWPREFS. Enlevé une partie idiote inutile obsolète de 8 octets dans le programmes de prefs...

Revision V2.01

-----  
Bug réparé: Injector provoquait un gourou si aucun fichier de préférences n'avait été créé... Désolé ! Les changements mineurs doivent être pensés minutieusement !

---

## 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 tooltips 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 tooltips. 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 affichée est finalement un gadget réellement sélectionnable, 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 maintenant 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 maintenant 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 ---

## 1.47 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 débbugger 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.

Quelques petites choses que j'inclurai peut-être dans la prochaine version:

- Une version MUI du programme de préférences (si j'arrive à la débbugger!)