

# ToolManager

---

Une extension pour le Workbench Amiga

Version 2.1  
16 Mai 1993

---

Copyright © 1990-93 Stefan Becker

La permission est accord'ee de faire et de distribuer des copies de ce manuel pourvu que la notice de copyright et cette notice de permissions soient preserv'ees sur toutes les copies.

Aucune garantie de quelque nature n'est donn'ee pour une fiabilit'e 'a 100% des programmes d'ecrits dans ce document. Vous utilisez votre mat'erieel 'a vos propres risques. L'auteur **ne peut pas**  tre rendu pour responsable des dommages caus'es par l'utilisation de ces programmes.

Cette archive est en libre distribution, mais reste encore "copyrighter" par Stefan Becker. Cela veut dire que vous pouvez la copier gratuitement aussi longtemps que vous ne demandez pas plus que le tarif nominal de copie. Ce tarif **ne doit pas**  tre plus de US \$5 ou 5 DM.

**Cette limite s'applique aux distributeurs de Domaine-Public Allemand aussi !!**

La permission est accord'ee pour inclure cette archive dans les collections de Domaine-Public, sp'ecialement dans les "Fred Fishs Amiga Disk Library" (en incluant les versions CD-ROM). Le fichier distribu'e peut  tre r'ecup'er'e sur des "Bulletin Board Systems" (BBS) ou sur des serveurs FTP. Si vous voulez distribuer ce programme, vous **devez** utiliser les archives de la distribution originale 'ToolManager2\_1bin.lha', 'ToolManager2\_1gfx.lha', 'ToolManager2\_1src.lha'.

Aucun de ces programmes ni le code source (ni des parties de celui-ci) ne peuvent  tre utilis'es dans des programmes commerciaux qu'avec la permission de l'auteur.

**Aucun** de ces programmes **ni** le code source (ni des parties de celui-ci) ne doivent  tre utilis'es sur aucune machine utilis'ee pour la recherche, le developpement, la construction, pour le test ou la production d'armes ou pour d'autres applications militaires. Cela inclut, 'egalement, toute machine qui est utilis'ee pour la formation des personnes pour **tous** les usages mentionn'es au dessus.

# 1 Important

Bienvenu dans le merveilleux monde de ToolManager 2.1 :-)

- ToolManager et ses concepts ont radicalement chang'e (voir Appendice B[Historique], page 31) depuis la dernière version (1.5).
- En commençant avec la version 2.0 de ToolManager, ce programme a une option *GiftWare*. Si vous aimez ce programme et que vous utilisez assez souvent, tu devras consid'erer l'envoi d'une petite donation pour honorer l'effort que l'auteur a mis dans ce programme. Je sugg'ere une donation de \$10-20 US ou 10-20 DM. SVP N'envoyer pas de ch'èques ou d'argent depuis l'exterieur de l'Europe, car le plus souvent, encaisser ces choses coûte plus que leur valeur.

Si vous n'envoyez pas votre contribution ou vos moyens ne le permettent pas, vous ne sentez pas g'en'e pour cela. Envoyez-moi un mot en me disant que vous utilisez ToolManager de toute façon. (J'aime avec du courrier de mes fans :-). Voir Chapitre 2 [Adresses des auteurs], page 2.

- Les utilisateurs de Toolmanager 1.X/2.0 peuvent commencer au chapitre d'installation rapide (voir Chapitre 3 [Installation rapide], page 3). Quelques caract'eristiques n'ont pas chang'e et le reste est 'equitablement facile 'a deviner 'a travers essais & erreurs. Pour une description du nouveau concept & caract'eristiques, consulter la partie r'ef'ERENCE de ce document (voir Chapitre 8 [Objets], page 12).

Vous **devez** quitter tout ToolManager 1.X/2.0 ou sinon la nouvelle version ne marchera pas. La nouvelle version ne peut pas lire l'ancien format du fichier de configuration 1.X (D'esol'e).

- Les utilisateurs d'ebutants doivent lire le document entier pour comprendre le concept et l'usage du programme. Commencer par Chapitre 4 [Introduction], page 4.
- ToolManager 2.1 utilise quelques caract'eristiques de l'AmigaOS Version V38 (et plus), qui supporte les nouvelles caract'eristiques r'eseau AmigaOS, qui devrait être (esp'érons-le) bientôt disponible pour tous les utilisateurs Amiga. Mais si vous utilisez encore le 2.0 (not'e V37 dans ce document), vous n'avez pas besoin d'être inquiet car ToolManager ne s'appuie pas sur ses caract'eristiques. Toutes les caract'eristiques 'etendues sont marqu'ees dans cette documentation.

## 2 O‘u envoyer la liste des erreurs, commentaires & donation

L’auteur peut être joint aux adresses suivantes :

Adresse postale :

Stefan Becker  
Holsteinstrasse 9  
5100 Aachen  
GERMANY

SVP, utiliser l’adresse suivante après le 1 Juillet 93:

Stefan Becker  
Holsteinstrasse 9  
52068 Aachen  
GERMANY

InterNet E-Mail:

`stefanb@pool.informatik.rwth-aachen.de`

## 3 Comment installer ToolManager 2.1 rapidement

L'installation basique de Toolmanager 2.1 se fait en quatre temps :

`'Libs/toolmanager.library' ⇒ 'LIBS:'`

C'est le programme principal de ToolManager. Il gère tous les programmes, menus, icônes et docks (voir Chapitre 10 [Librairie], page 23).

`'Prefs/ToolManager*' ⇒ 'SYS:Prefs'`

C'est l'éditeur pour la configuration (voir Chapitre 9 [Preferences], page 19).

`'WBStartup/ToolManager*' ⇒ 'SYS:WBStartup'`

Avec cet outil, vous pouvez démarrer ou arrêter ToolManager. Si il réside dans le répertoire WBStartup, ToolManager sera toujours lancé quand votre machine bootera.

`'L/WBStart-Handler' ⇒ 'L:'`

Ce programme lance les programmes par le Workbench. C'est un processus séparé, de ce fait, vous pouvez quitter ToolManager même si vous avez encore des programmes actifs lancés par ToolManager par la méthode WB.

Après avoir copier ces fichiers, vous devrez quitter toute ancienne version de ToolManager et double-cliquer l'icône dans le répertoire WBStartup. Maintenant vous pouvez lancer l'éditeur de préférences et configurer votre environnement (Utilisez le bouton "Test" au lieu du bouton "Use (Utilise)" pendant les tests). Vous devriez comprendre la plus part des caractéristiques avec des essais & des erreurs, pour plus d'informations : voir les descriptions des objets de ToolManager (voir Chapitre 8 [Objets], page 12).

La distribution inclut un exemple de configuration appelé `'TM.Demo.prefs'`. Vous pouvez le charger dans l'éditeur de préférences avec l'entrée du menu **Open/Ouvre**.

## 4 Qu'est ToolManager ?

ToolManager est un programme souple pour g'erer les outils de votre environnement de travail. Il peut lancer des programmes du Workbench et du CLI, des scripts AREXX, des 'ev'ènements sur des raccourci-clavier. Il peut même lancer une commande d'un ToolManager tournant sur une machine 'a distance. L'interface utilisateur est consistu'ee de menus, d'icônes ou de fenêtres dock. Si vous aimez un ordinateur bruyant, vous pouvez associer un son 'a chacun de ces objets. Voir Paragraphe 8.3 [Sound], page 14.

ToolManager peut rajouter des objets dans le menu **Tools** du Workbench. Si vous selectionnez un objet du menu, le programme associ'e avec lui sera lanc'e. Chaque icône selectionn'ee sur le Workbench sera utilis'ee comme un argument pour le programme. Cette caract'eristique est seulement disponible quand le Workbench est actif. Voir Paragraphe 8.4 [Menu], page 14.

ToolManager peut rajouter des icônes 'a la fenêtre du WorkBench. Quand vous double-cliquez une pareille icône, le programme associ'e avec elle sera lanc'e. Si vous amenez une ou plusieurs icônes sur cette icône, le programme sera lanc'e avec ces icônes comme arguments. Cette caract'eristique est seulement disponible quand le Workbench est actif. Voir Paragraphe 8.5 [Icon], page 15.

ToolManager peut cr'eer une fenêtre dock depuis un regroupement de programmes. Cette fenêtre peut être ouverte sur chaque 'ecran public. Chaque programme est represent'e par une image ou un bouton gadget. Pour lancer un programme, vous cliquez simplement sur l'image ou le bouton gadget. Si la fenêtre dock a 'et'e ouverte sur l''ecran du Workbench et qu'il est actif, vous pouvez aussi amener une ou plusieurs icônes sur l'image ou le bouton gadget pour lancer le programme avec les arguments. Voir Paragraphe 8.6 [Dock], page 16.

En plus, vous pouvez associer une touche d'appel 'a chaque programme. Si vous pressez cette Hot Key, le programme sera lanc'e. A noter que *aucun* argument ne peut être pass'e au programme si vous utilisez cette m'ethode de lancement. Voir Paragraphe 8.1 [Exec], page 12.

## 5 Les concepts de ToolManager

ToolManager 2.1 utilise une nouvelle approche orient'ee-objet pour fournir un syst'eme souple et extensible. Cette approche accroît plusieurs caract'eristiques des versions 1.X de ToolManager, e.g. vous pouvez avoir, maintenant, des docks multiples.

Un objet est un regroupement de donn'ees qui d'ecrivent ces caract'eristiques. Chaque objet a un nom et un type. Vous pouvez cr'eer autant d'objets que vous voulez, mais le nom de chaque objet doit être unique, parce qu'il sera utilis'e comme une r'ef'erence pour cet objet.

Actuellement, il y a sept diff'erents types d'objets: Exec, Image, Sound, Menu, Icon, Dock et Access. Les trois premiers sont des objets primaires; ce qui veut dire qu'ils ne r'ef'erencent pas d'autres objets. Ils fournissent les donn'ees ou services pour les objets complexes.

Les quatre derniers types d'objets sont des objets complexes; ce qui veut dire qu'ils r'ef'erencent des objets simples et reposent sur eux l'acc'es aux donn'ees ou services. La r'ef'erence est faite par nom, et si aucun objet avec ce nom existe, l'objet complexe l'ignorera. A noter que cela peut reduire la fonctionnalit'e de l'objet complexe, e.g un objet Icon a *besoin* de la donn'ee d'un objet Image, et si cet objet n'existe pas, une icône ne sera pas cr'ee'e.

Pour une description d'etaill'ee de tous les param'etres voir Chapitre 8 [Objets], page 12.

## 6 Un tour guidé à travers ToolManager

Si vous n'avez pas compris un mot jusqu'à présent ? Confus par les objets, programmes et les liens ? Ne désespérez pas, ce qui suit va t'aider.

Je voudrais maintenant te guider à travers un exemple pas-à-pas sur comment configurer ToolManager. Tout ce dont vous avez besoin est d'installer ToolManager et de lancer l'éditeur de préférences. Après chaque pas, utilise le bouton "Test" dans la fenêtre principale pour tester la configuration.

Comme exemple, nous allons prendre l'afficheur de texte More du tiroir 'SYS:Utilities'. D'abord, nous devons dire à ToolManager quel programme nous voulons utiliser. L'information sur les programmes est stockée dans Objets Exec. Sélectionne juste "Exec" comme type d'objet dans la fenêtre principale de l'éditeur de préférences et presse le bouton "Créer".

Après cette action, tu verras la fenêtre "Modifier l'objet Exec". Maintenant, ouvre le tiroir Utilities de ta partition Workbench, déplace l'icône More pour l'amener dans la fenêtre d'édition. Comme tu peux le voir, l'éditeur a maintenant mis le nom de l'objet et la commande au nom de programme, et le répertoire courant à System:Utilities. Presse le bouton "Ok" pour utiliser ces paramètres.

Tu ne peux pas faire grand chose avec un objet Exec seul, aussi, comme prochain pas, nous voulons ajouter ce programme au menu Outils du Workbench. Sélectionne "Menu" comme type d'objet et presse le bouton "Créer". Maintenant tu verras la fenêtre "Modifier objet Menu". Change le nom de l'objet à "Affichage Texte".

ToolManager doit savoir quel programme doit être lancé quand l'entrée du menu est sélectionnée, aussi nous associons un objet Exec à l'objet du menu. Presse le bouton "Objet Exec" et sélectionne l'objet "More" depuis la boîte de dialogue. Maintenant, presse le bouton "Ok" et le bouton "Test" dans la fenêtre principale. Vous pouvez, maintenant, voir une entrée dans le menu "Outils". Sélectionne un fichier texte depuis le Workbench et choisis la nouvelle entrée du menu. Le programme "More" sera lancé et affichera le texte. C'est facile, non ?

Maintenant, nous allons créer un objet Icon sur le Workbench. Pour une icône, nous avons besoin d'une image, qui est stockée dans un objet Image. Choisis "Image" comme nouveau type d'objet et presse le bouton "Créer". La fenêtre "Modification objet Image" sera ouverte. Change le nom à "Image pour More" et amène l'icône de More depuis le tiroir Utilities sur la fenêtre. Presse "Ok" pour utiliser ces paramètres.



Ensuite, nous allons créer un objet Icône. Choisis “Icon” comme type d’objet et presse le bouton “Créer”. Change le nom de l’objet à “Voir Texte”. Presse le bouton “Objet Exec” et choisis l’objet “Image pour More” depuis la boîte de dialogue. Mets la position X à 100 et la position Y à 50. Presse le bouton “Ok” et le bouton “Test”. Après un bref delay, une icône apparaîtra dans le Workbench, sur laquelle tu pourras amener des icônes de tes fichiers texte pour les afficher.

Je suis sûr que, maintenant, tu as une idée comment utiliser les objets ToolManager et de la façon de lier entre eux pour construire ton environnement. Maintenant, tu peux tester le reste des caractéristiques en essayant une à une. Tu peux aussi regarder la configuration d’emo dans le fichier ‘TM\_Demo.prefs’

## 7 Description de tous les fichiers de la distribution

La distribution de ToolManager 2.1 est consistée de plusieurs répertoires qui sont expliqués en dessous. A noter que la distribution est découpée en trois parties, aussi, il se peut que vous n'ayez pas tous les répertoires qui sont mentionnés en dessous.

### 7.1 Le répertoire de documentations

Ce répertoire contient la documentation de ToolManager. La documentation est disponible dans quatre formats différents et dans plusieurs langues. En plus, il y a un fichier au format AutoDoc décrivant l'interface de la librairie partagée de ToolManager.

Préfixe 'TM\_<language>'

Ce fichier contient la documentation pour le langage spécifié. Actuellement les langages disponibles sont : l'Allemand, l'Anglais, le Français, le Svenska.

Postfixe '.doc'

Ce fichier contient la documentation sous forme d'un texte ASCII.

Postfixe '.dvi'

Ce fichier contient la documentation en format TeXs DVI. Pour avoir un manuel imprimé, utilisez ce fichier avec une commande pour imprimer du TeX.

Postfixe '.guide'

Ce fichier contient la documentation au format AmigaGuide. Bien que ce soit seulement de l'ASCII avec quelques commandes, vous avez besoin d'AmigaGuide pour exploiter les liens hypertexte dedans.

Postfixe '.tex'

Ce fichier contient la documentation au format Texinfo, comme spécifié par la Free Software Foundation (FSF). Avec le paquetage de macros dans 'texinfo.tex', vous pouvez utiliser, ensemble, TeX et 'texindex' pour créer un fichier au format DVI (voir au-dessus).

'toolmanager.doc'

Ce fichier contient la description de l'interface de la librairie partagée au format AutoDoc.

### 7.2 Le répertoire de Goodies

Ce répertoire contient des archives de programmes additionnels qui sont très pratiques pour les

opérations de ToolManager.

**‘GetPubName.lha’**

Ce petit programme donne le nom de l’écran public le plus en avant soit sur stdout ou dans une variable d’environnement. Il a été écrit par Michael “Mick” Hohmann.

**‘upd1\_20.lha’**

Le programme **upd** a été écrit par Jonas Petersson. C’est un petit programme qui ouvre un port ARexx et attend des commandes. Via ARexx, vous pouvez ordonner **upd** pour jouer des fichiers son. ToolManager utilise cette caractéristique pour implémenter ses objets Sound. Voir Paragraphe 8.3 [Sound], page 14.

## 7.3 Le répertoire Graphics

Ce répertoire contient une riche collection d’images parmi celles que vous avez choisies comme vos favorites. Chargez-les comme des objets Images dans ToolManager (voir Paragraphe 8.2 [Image], page 13).

Les fichiers sont contribués par diverses personnes (voir Appendice C[Credits], page 32). Chacun d’eux a un sous-répertoire séparé dans la distribution. Comme les fichiers ont été créés par différents auteurs, ils viennent d’environnements différents (palette, profondeur, résolution, taille) et ont des styles différents. Ainsi, pas toutes les images peuvent ne pas avoir l’air bonnes pour votre machine.

Pour différencier les formats d’images qui sont supportés par ToolManager, chaque fichier a un suffixe qui décrit le format du fichier :

- ‘.anmb’** C’est un fichier IFF ANIM créé par un programme de dessin/animations. Il contient plusieurs images. Bien que ToolManager peut charger des fichiers ANIM en entier, vous devez utiliser quelque chose comme des “AnimBrush” DPaint pour découper la partie intéressante de l’animation.
- ‘.brush’** C’est un fichier IFF ILBM créé par un programme de dessin. Il contient seulement une image.
- ‘.info’** C’est une icône normale Amiga créée avec IconEdit (ou quelque chose de similaire). Elle contient deux images.

## 7.4 Le répertoire L

Ce répertoire contient seulement un fichier, nommé `WBStart-Handler`. Vous devez copier ce fichier dans le répertoire `L:`, ou sinon ToolManager ne sera pas capable de démarrer un objet Exec par la méthode du WB (voir Paragraphe 8.1 [Exec], page 12).

Le paquetage complet WBstart 1.2 peut être trouvé sur une Disquette Fish #757

## 7.5 Le répertoire Libs

Ce programme contient seulement un fichier, `toolmanager.library`. C'est le programme principal pour ToolManager et doit être copié dans le répertoire `LIBS:`.

## 7.6 Le répertoire Locale

Ce répertoire contient tous les fichiers pour le support Local de ToolManager. Comme `locale.library` est nouvelle avec V38, vous n'avez pas besoin de ces fichiers si vous utilisez V37. Si vous utilisez V38, choisissez les fichiers pour votre langue et copier-les aux endroits appropriées.

`'Catalogs/<language>/toolmanager.catalog'`

C'est un fichier de traduction pour un langage spécifique. Copiez le fichier pour votre langue dans le répertoire `'LOCALE:Catalogs/<language>'`.

`'Languages/<language>.language'`

Quelques langages ne sont pas fournis par la distribution standard Locale V38. Alors certains traducteurs ont rajouté un fichier `language`, alors ToolManager peut utiliser ces fichiers de traductions. Copiez le fichier pour votre langue dans le répertoire `'LOCALE:Languages'`. Les langages additionnels disponibles sont: le Finlandais (suomi), Eefeler Platt (eiffel).

## 7.7 Le répertoire Prefs

L'éditeur de préférences de ToolManager et son icône résident dans ce répertoire. Copiez ces deux fichiers dans le répertoire `'SYS:Prefs'`. Pour plus d'informations sur l'éditeur voir Chapitre 9 [Preferences], page 19.

## 7.8 Le r pertoire Programmers

Ce r pertoire contient tous les fichiers qui sont n cessaires pour les diff rents langages et leurs compilateurs pour utiliser l'interface de la librairie partag e de ToolManager. Regardez dans le sous-r pertoire '**examples**' pour quelques exemples sur comment utiliser cette interface. Pour une description compl te de l'interface, lire le fichier '**Docs/toolmanager.doc**'.

Actuellement, les langages/compilateurs support es sont: AmigaOberon, DICE C, M2Amiga Modula-2, MANX Aztec C et SAS C.

## 7.9 Le r pertoire Scripts

Ce repertoire contient une collection de scripts ARexx ou Shell qui peuvent  tre utilis es comme objets Exec dans ToolManager. A noter qu'ils peuvent  tre sp cifiques 'a l'environnement ou shell des auteurs, alors il se peut que vous devez les modifier.

## 7.10 Le r pertoire Source

Ce r pertoire contient le code source complet de ToolManager 2.1 et de ses utilitaires. Chaque programme a son propre sous-r pertoire. L'auteur fournit le code source comme un exemple pour la programmation OS 2.x/3.0.

Le sous-r pertoire '**locale**' a de l'inter t pour les traducteurs. Si votre langage n'est pas fourni dans cette version et vous voulez faire la traduction, regardez le fichier '**empty.ct**'. Remplissez juste les lignes vides et envoyez-moi le fichier. Il se peut qu'il soit inclus dans la prochaine version.

## 7.11 Le r pertore WBStartup

Seulement un programme r side dans ce r pertoire: **ToolManager**. Cet utilitaire d'emarre et arr te ToolManager 2.1. La plupart du temps, cet utilitaire sera dans le r pertoire '**SYS:WBStartup**', mais il peut  tre utilis e depuis le Shell aussi.

## 8 R  f  rence des objets ToolManager

Ce chapitre d  crit les objets de ToolManager dans le d  tail. Chaque objet a un type et un nom. Le nom est utilis   pour r  f  rencer l'objet. Il y a six diff  rents types d'objets:

### 8.1 Exec objets

Les objets Exec d  crivent les programmes qui vont   tre lanc  s par ToolManager. Trois diff  rents types de programmes sont support  s: CLI, Workbench et ARexx. Les trois types d'actions sont support  s : Dock, Hot Key, Network. Chaque objet Exec a les param  tres suivants<sup>1</sup>. Les valeurs par d  faut sont mises entre parenth  ses:

#### Arguments/Param  tres (Yes)

Cet interrupteur controle la transmission des param  tres au programme. Si un programme ne supporte pas les param  tres ou n'en a pas besoin, vous pouvez mettre sur off le passage de param  tres.

#### Command/Commande

Le nom de fichier du programme   a lancer. Ce nom peut   tre relatif au catalogue courant. Si le type est Dock, la commande d  crit le nom de l'objet dock, qui doit   tre ouvert/ferm  . Pour le type Hot Key, cette cha  ne doit   tre une Cha  ne de Description d'Entr  e Commodit  s (voir Chapitre 11 [Raccourci-Clavier], page 24). Une commande distante (type Network) est d  crite comme **objet@machine**, qui dit   a ToolManager de lancer sur la **machine** l'objet Exec nomm   **objet**.

#### Current Directory/Catalogue courant ('SYS:')

Le nom du catalogue courant pour le programme. Note: les programmes ARexx ignorent ce param  tre.

#### Delay/D  lai (0)

Apr  s activation d'un objet Exec, ToolManager attend pendant D  lai secondes avant de lancer le programme. Si cette valeur est n  gative, le programme sera lanc   chaque D  lai secondes. Pour stopper un objet Exec qui attend son ex  cution, activez-le juste une fois encore. Note: Si D  lai est positionn  , le programme sera lanc   sans arguments.

---

<sup>1</sup> Pour des raisons de traductions et des versions de Workbench V37/V38, les noms des objets sont en Anglais/Fran  ais

**Exec Type/Mode Exec (CLI)**

Cela spécifie le mode du programme. Il peut être un parmi cela: CLI, WB ou ARexx.

**Hot Key/Raccourci-clavier**

Vous pouvez mettre une touche d'appel pour chaque objet Exec. Si cette touche est pressée, le programme sera lancé. Note: Le programme sera lancé sans arguments.

**Output File/Fichier de sortie ('NIL:')**

C'est le nom du fichier pour le fichier de sortie. Il est seulement utile pour les programmes CLI.

**Path/Chemin (chemin pour les processus**

ToolManager) Cette chaîne positionne le chemin de recherche pour le programme. Vous pouvez spécifier plusieurs répertoires en séparant les noms par des “;”. C'est seulement utile pour les programmes CLI.

**Priority/Priorité (0)**

Cela positionne la priorité du nouveau processus qui lancera le programme.

**Public Screen/Ecran public (d'efaut 'ecran public)**

Vous pouvez positionner le nom de l'écran public qui devra être déplacé vers l'avant avant que le programme soit lancé. Cela marche seulement en conjonction avec le paramètre **To Front/Au premier plan**.

**Stack/Pile (4096)**

Ceci positionne la pile d'exécution du nouveau processus qui lance le programme.

**To Front/Au premier plan (No)**

Si vous positionnez ce paramètre, l'écran public spécifié par **Public Screen/Ecran public** est déplacé au premier plan avant que le programme soit lancé.

## 8.2 Objets Image

Les objets Image spécifient l'image qui va être utilisée par ToolManager pour les icônes ou les docks. Ce type d'objet a seulement un paramètre:

**File Name/Nom du fichier**

Cela spécifie le nom du fichier depuis lequel ToolManager devra lire les données images. ToolManager essaye de détecter le type de l'image automatiquement:

1. Il essaye de la charger comme une image IFF. Actuellement, ToolManager peut lire les fichiers ILBM (une image) ou ANIM (deux ou plusieurs images).
2. Il essaye de la lire comme un fichier icône. Une icône peut avoir un ou deux images.

Les animations sont actuellement supportées seulement par les objets Dock. Les objets Icon

comportent seulement la première et la seconde image d'une animation pour construire une icône à deux images. Si vous voulez une animation pour ToolManager, vous devez suivre ces règles de dessin:

- Image 1      Cela doit être une image qui représente un 'etat inactif.
- Image 2      Cela doit être une image qui représente un 'etat s'electionn'e. Normalement, c'est une copie invers'ee de la première image.
- Image 3 to N-1  
                  Ce sont des images pour l'animation. Chaque image sera affich'ee pendant 1/3 de seconde.
- Image N      Cette dernière image de l'animation sera affich'ee pendant une seconde. Apr'es celle-la, la première sera de nouveau affich'ee.

## 8.3 Sound objets

Un objet Sound peut être utilis'e pour rendre ToolManager bruyant. ToolManager lui-même a aucune capacit'e interne pour jouer des sons, il utilise ARexx pour activer un d'emon externe jouant des sons. Ce type d'objet a deux param'etres:

### Command/Commande

Cela positionne la commande ARexx que ToolManager envoie pour envoyer un joueur de sons externe. Pour **upd**, cela pourrait être quelque chose comme **file samples:boing**. Voir Paragraphe 7.2 [Goodies], page 8.

### Port ARexx

Cela sp'ecifie le port ARexx o'ù ToolManager doit envoyer la **commande**. La valeur par d'efaut est **PLAY** qui est le port pour le programme **upd**.

## 8.4 Menu objets

Les objets Menu controlent les entr'ees dans le menu Outils/Tools du Workbench. Le nom de l'objet est utilis'e comme le texte du menu. Pour activer un pareil objet, s'electionner juste le nom dans le menu. Les objets Menu marchent seulement quand le Workbench tourne.

Ce type d'objet a deux param'etres:



**Exec Objet**

C'est le nom d'un objet Exec qui doit être activé quand l'entrée dans le menu est sélectionnée. Chaque icône qui est sélectionnée à ce moment sera utilisée comme un argument pour le programme.

**Sound Objet**

C'est le nom de l'objet Sound qui doit être activé quand l'entrée du menu est sélectionnée. ■

A noter pour les utilisateurs de ToolManager 1.X: Pour simuler le vieux outil de type "Dummy" créer juste un objet Menu et en spécifiant *aucun* objet Sound et Exec.

## 8.5 Icon objets

Les objets Icon écrivent les icônes dans la fenêtre Workbench. Un objet peut être aussi bien activé en double-cliquant l'icône ou en amenant une icône sur l'icône d'application. Les objets Icon ne fonctionnent seulement si le Workbench tourne.

Les paramètres pour ce type d'objet sont les suivants:

**Exec Objet**

C'est le nom d'un objet Exec qui devra être activé quand l'icône est sélectionnée. Chaque icône qui est amenée sur l'icône d'application sera utilisée comme argument pour le programme.

**Image Objet**

C'est le nom d'un objet Image. Le nom de cet objet est utilisé pour construire l'icône d'application.

**Left Edge/Bord Gauche** (défaut: 0)

Cela positionne le bord gauche pour l'icône d'application.

**Show Name/Montrer le nom** (défaut: Oui)

Si ce paramètre est mis, le nom de l'objet sera utilisé comme nom pour l'icône d'application.

**Sound Objet**

C'est le nom d'un objet Son qui devra être activé quand l'icône est activée.

**Top Edge/Bord supérieur** (défaut: 0)

Cela positionne le bord supérieur pour l'icône d'application.

Note: Le Workbench est *très* pointu à propos de la position des icônes. Si vous spécifiez des coordonnées que le Workbench n'aime pas, il les ignorera et placera l'icône ailleurs.

## 8.6 Dock objets

Les objets Dock d'écrivent des fenêtres. Ces fenêtres combinent plusieurs outils qui sont représentés par des images ou gadgets. Pour lancer un outil, cliquer juste sur son image ou gadget. Bien sûr, vous pouvez amener des icônes sur l'image ou le gadget pour le passer en argument pour l'outil.

Chaque objet dock a plusieurs paramètres. Les valeurs par défaut sont mis entre parenthèses:

### **Activated/Activée (Oui)**

Une fenêtre dock peut être active (ouverte) ou inactive (fermée).

### **Backdrop/Arrière-plan (Non)**

La fenêtre Dock doit aller immédiatement en arrière-plan après l'ouverture.

### **Centered/Centree (Non)**

Si ce paramètre est positionné, la fenêtre sera toujours centrée par rapport à la position courante de la souris à son ouverture.

### **Columns/Colonnes (1)**

Ce paramètre détermine le nombre de colonnes d'outils. Les outils sont toujours classés par colonne, en commençant par la colonne la plus à gauche et en remplissant vers le haut en les classant vers la colonne la plus à droite.

### **Font/Police (Police d'écran)**

Si vous avez une fenêtre dock avec le paramètre **Texte** mis, vous pouvez choisir la police pour les gadgets boutons avec ce paramètre.

### **Frontmost/Premier plan (Non)**

Si vous mettez ce paramètre, la fenêtre dock sera toujours ouverte sur l'écran public le plus en avant.

### **Hot Key/Raccourci-clavier**

Vous pouvez mettre un Raccourci-clavier pour chaque objet Dock. Si ce Raccourci-clavier est généré, l'état d'activation de la fenêtre dock sera basculé; avec cela, elle sera fermée ou ouverte.

### **Left Edge/Bord gauche (0)**

Ce paramètre positionne le bord gauche de la fenêtre dock. Si le paramètre **Centré** est mis, ce paramètre sera ignoré.

### **Menu (Non)**

Vous pouvez ajouter un petit menu. Ce menu a deux entrées:

**Close Dock/Fermer Dock** Ferme la fenêtre Dock

**Quit TM** Quitte ToolManager

### **Pattern/Fond (Non)**

La fenêtre dock ajuste automatiquement sa taille à l'image la plus large. Chaque entrée de la dock a la même taille, et les plus petites images sont centrées, alors elles ont une bordure transparente autour d'elles. Si vous n'aimez pas cette bordure transparente, mettez ce paramètre et la bordure sera remplie avec un motif.

#### PopUp/Sur demande (Non)

Quand ce paramètre est mis, la fenêtre dock sera automatiquement fermée après la sélection d'une entrée dans la Dock. Cela est spécialement pratique avec les paramètres **Centered/Centrée**, **Frontmost/Premier plan** et un raccourci-clavier de la classe **rawmouse** (voir Chapitre 11 [Raccourci-Clavier], page 24).

#### Public Screen/Ecran public (Par défaut 'écran public)

Spécifie l'écran public sur lequel la fenêtre dock doit être ouverte. Si la fenêtre dock a été ouverte avec un raccourci-clavier, l'écran public sera d'éplacé en avant après que la fenêtre eut été ouverte. Ce paramètre sera ignoré si le paramètre **Frontmost/Premier plan** est activé.

#### Sticky/Collée

Normalement, une fenêtre Dock garde sa dernière position quand vous la fermez et apparaît à la même position quand vous la ré-ouvrez. Si vous voulez que la fenêtre Dock s'ouvre toujours à la même position, vous devez mettre ce paramètre.

#### Texte (Non)

Vous pouvez choisir entre des gadgets images ou boutons dans les fenêtres dock avec ce paramètre. Les fenêtres dock à boutons sont spécialement pratiques avec le paramètre **PopUp**.

#### Title/Titre

Cela spécifie le titre de la fenêtre dock. Si vous mettez un titre, la fenêtre dock sera une fenêtre normale sous OS 2.0 avec une barre de déplacement, un gadget de fermeture, un gadget de profondeur, et une bordure. Si vous *ne mettez pas* de titre, vous aurez une fenêtre dock avec seulement une barre de déplacement et *sans* bordure.

#### Top Edge/Bord supérieur (0)

Ce paramètre positionne le bord supérieur de la fenêtre dock. Si le paramètre **Centered/Centrée** est mis, ce paramètre sera ignoré.

#### Vertical (Non)

Si la fenêtre dock a un style de nouvelle fenêtre (i.e. : seulement une barre de déplacement et aucune bordure), ce paramètre positionne l'orientation de la barre de déplacement. Ce paramètre est ignoré si vous mettez un titre à la fenêtre avec le paramètre **Title/Titre**.

## 8.7 Access objets

Les objets Access contrôlent les droits d'accès pour les demandes réseau. Par défaut, *chaque* requête est refusée, aussi un ToolManager distant ne peut satisfaire l'opération de votre machine qu'en activant vos objets Exec. Avec des objets Access, vous pouvez permettre aux machines spécifiées d'activer vos objets Exec. Le nom d'un objet Access a une signification spéciale. Il est constitué avec le nom de la machine distante depuis laquelle la requête réseau a été envoyée. ToolManager utilise les trois étapes suivantes :

1. Complète avec le nom complet de la machine hôte
2. Complète avec le nom réel
3. Regarde pour l'objet Access appelé **anyone**

Si l'objet correspondant est trouvé, alors cet objet est utilisé pour les droits d'accès de la machine distante. L'objet appelé **anyone** est utilisé pour n'importe quelle requête réseau, pour laquelle l'objet Access correspondant ne peut être trouvé.

L'objet Access a seulement un paramètre :

### Exec Objet

Ce paramètre peut être utilisé plusieurs fois et spécifie quels objets Exec peuvent être activés depuis la machine. Si vous n'avez spécifié *aucun* nom d'objet, alors la machine distante peut activer *tous* les objets Exec sur votre machine.

## 9 L'éditeur de préférences de ToolManager

Avec l'éditeur de préférences, vous pouvez gérer la configuration globale de ToolManager. Cette configuration est chargée automatiquement quand vous lancez ToolManager. Pour lancer l'éditeur, double-cliquer juste son icône. Vous verrez la fenêtre principale.

La plupart des gadgets dans les fenêtres de l'éditeur ont un raccourci-clavier. Ils sont marqués par un souligné (\_). À noter que si une chaîne gadget est active, vous devez d'abord presser la touche return avant d'utiliser les raccourci-clavier.

### 9.1 Fenêtre principale Gadgets

La fenêtre principale a plusieurs groupes de gadgets:

#### Type d'objet

Avec un gadget cyclique, vous pouvez choisir le type d'objets que vous voulez créer ou éditer.

#### Liste d'objets

Ce gadget montre la liste de tous les objets du type courant. Vous pouvez sélectionner un objet en cliquant sur son nom. Si vous double-cliquez un objet, une fenêtre d'édition sera ouverte.

#### Déplacement d'objets

Quand un objet est sélectionné, vous pouvez le déplacer dans la liste avec ces gadgets. Si vous cliquez sur le gadget **Sort/Trier**, les objets dans la liste seront classés par ordre alphabétique.

#### Manipulation d'objets

Ces gadgets manipulent les objets. Le gadget **New/Créer** crée un nouveau objet du type courant qui sera sélectionné automatiquement. Quand vous cliquez sur le gadget **Edit/Modifier**, une fenêtre d'édition pour l'objet sélectionné sera ouverte. Avec le gadget **Copy/Copier**, vous pouvez faire une copie de l'objet sélectionné. Le gadget **Remove/Supprimer** efface l'objet sélectionné.

#### Configuration

Vous avez plusieurs choix pour sauver la configuration. Avec le gadget **Save/Sauver**, vous pouvez sauver la configuration de façon permanente dans le fichier `'ENVARC:ToolManager.prefs'`. Pour une modification temporaire, utilisez le gadget **Use/Utiliser**, qui sauvera la configuration dans le fichier `'ENV:ToolManager.prefs'`. Ce fichier ne survivra pas à un

reset de la machine. Pour tester la nouvelle configuration sans quitter l'éditeur, utilisez le gadget **Test/Tester**. Le gadget **Cancel/Annuler** quittera l'éditeur sans sauvegarde.

## 9.2 Menus fenêtre princiale

La fenêtre principale a plusieurs menus:

### Project/Projet

Avec les sous-menus **Open/Ouvrir** et **Save as/Sauver comme**, vous pouvez charger et sauver la configuration. Le sous-menu **About/A propos** ouvre une requête d'information. La s'el'ection du sous-menu **Quit/Quitter** quittera l'éditeur sans sauvegarde.

### Edit/Modifier

Avec ces sous-menus, vous pouvez rappeler des configurations plus vieilles. Le sous-menu **Last Saved/Reprendre 'etat sauvegard'e** charge la dernière configuration sauvee du fichier 'ENVARC:ToolManager.prefs'. Avec le sous-menu **Restore/Reprendre valeurs initiales**, vous chargerez la configuration, qui 'etait active avant d'ouvrir l'éditeur, du fichier 'ENV:ToolManager.prefs'.

### Settings/Options

Vous pouvez choisir le sous-menu **Create Icons/Cr'eer les icônes** pour que le sous-menu **Save as/Sauver comme** cr'ee ou non une icône.

## 9.3 Fenêtres de modification

Chaque type d'objet a une fenêtre différente pour positionner les paramètres des objets. Pour une liste détaillée de tous les paramètres voir Chapitre 8 [Objets], page 12.

Chaque fenêtre d'edition a un gadget texte pour le nom de l'objet. Ce nom est important, parce qu'il est utilis'e pour r'ef'erençer cet objet. A noter qu'il n'y a actuellement pas de r'ef'ERENCE interne crois'ee. Alors si vous changez le nom d'un objet qui est d'ej'a r'ef'ERENC'e par un autre objet, cette r'ef'ERENCE *ne sera pas* mise 'a jour. Vous devez le faire 'a la main.

Les gadgets bouton dans les fenêtres d'edition ouvrent différents types de boîtes de dialogue. Vous pouvez choisir un objet en cliquant dessus et en pressant le gadget **Ok**, ou simplement en double-cliquant dessus. Pour quitter une boîte de dialogue sans modifications, utilisez le gadget **Cancel/Annuler**. Si vous voulez effacer un champs qui ne peut être seulement choisi par une boîte de dialogue, ouvrez-la et presser le gadget **Ok** *sans* s'el'ectionner l'objet.

Les fenêtres d'édition pour les objets de type Exec et Image ont des caractéristiques supplémentaires. ■  
 Vous pouvez simplement amener une icône dessus pour passer les paramètres de cette icône.

## 9.4 Fenêtre de création d'objets.

Si vous amenez une icône sur la fenêtre principale, la fenêtre "Création d'objets" sera ouverte. Ici, vous pouvez choisir que les objets soient créés depuis cette icône. Cela peut être utilisé pour ajouter un programme à votre configuration très facilement et très rapidement.

Vous pouvez juste créer un objet Exec ou Image depuis une icône, si vous sélectionnez un des premiers choix. Mais, vous pouvez aussi créer un objet complet Menu et/ou Icon, si vous sélectionnez un des trois derniers choix.

## 9.5 Types d'outils

Quand vous lancez l'éditeur de préférences depuis le Workbench, vous pouvez positionner plusieurs types d'outils (tooltypes) dans l'information de l'icône ou le fichier de configuration de l'icône pour la contrôler.

USE	Si vous positionnez ce type d'outils dans une icône pour un fichier de préférences, l'éditeur installera ce fichier comme le fichier de configuration courante.
SAVE	Si vous positionnez ce type d'outils dans une icône pour un fichier de préférences, l'éditeur installera ce fichier comme le fichier de configuration courante et permanente.
PUBSCREEN	Ce type d'outils dit à l'éditeur d'ouvrir sa fenêtre sur un écran public spécifique. Si vous n'invoquez pas ce type d'outils, l'écran public par défaut sera utilisé.
CREATEICONS	Quand ce type d'outils est mis à YES, l'éditeur créera une icône pour chaque fichier de préférences qui est créé avec le sous-menu <b>Save As/ Sauver comme</b> .
DEFAULTFONT	L'éditeur normalement utilise la police de caractères de l'écran public pour dessiner ses gadgets. Si vous mettez ce type d'outil à YES, l'éditeur utilisera la police du système par défaut.
XPOS	Cela spécifie la position X initiale de la fenêtre principale de l'éditeur.
YPOS	Cela spécifie la position Y initiale de la fenêtre principale de l'éditeur.

**MINLISTCOLUMNS**

Cela spécifie le nombre minimum de colonnes dans la liste des gadgets.

**MINLISTROWS**

Cela spécifie le nombre minimum de lignes dans la liste des gadgets.

## 9.6 Arguments CLI

Quand l'éditeur est lancé depuis le shell, il utilise la ligne de commande suivante:

```
FROM,EDIT/S,USE/S,SAVE/S,PUBSCREEN/K,DEFAULTFONT/S
```

- FROM** Ce paramètre spécifie le nom du fichier de préférences que l'éditeur devra charger.
- USE** Si vous utilisez ce paramètre, l'éditeur installera le fichier spécifié comme le paramètre **FROM** comme le fichier courant de configuration.
- SAVE** Si vous utilisez ce paramètre, l'éditeur installera le fichier spécifié comme **FROM** comme le fichier courant et permanent de configuration.
- PUBSCREEN** Ce paramètre dit 'a l'éditeur d'ouvrir ses fenêtres sur un écran public spécifique. Si vous ne positionnez pas ce type d'outil, l'écran public par défaut sera utilisé.
- DEFAULTFONT** L'éditeur utilise normalement la police de caractères de l'écran public pour dessiner ses gadgets. Si vous utilisez ce paramètre, l'éditeur utilisera la police de caractères système par défaut.



## 10 L'interface de la librairie partag'ee de ToolManager

Le gestionnaire de ToolManager est mis dans une librairie partag'ee Amiga. Cette librairie offre plusieurs fonctions pour cr'eer et manipuler les objets ToolManager, ainsi vous pouvez les utiliser dans vos programmes.

Il y a actuellement six fonctions disponibles:

### **AllocTMHandle()**

Pour cr'eer des objets ToolManager, vous devez d'abord allouer un TMHandle. Ce gestionnaire stocke toutes les informations de vos objets et est utilis'e pour les r'eferencier. A noter que l'information stock'ee dans ce gestionnaire est *seulement* accessible par le programme qui les a cr'e'es.

### **FreeTMHandle()**

Cette fonction lib'ere un TMHandle et tous les objets ToolManager associ'es avec lui. Chaque AllocTMHandle() doit ˆetre termin'e par un FreeTMHandle()!

### **CreateTMOBJECTTags()**

### **CreateTMOBJECTTagList()**

Cette fonction cr'ee un objet ToolManager. Vous devez fournir un nom, le type d'objet et diff'erents renseignements pour les param'etres de l'objet. Le nom de l'objet est important, car il est utilis'e pour r'eferencer l'objet.

### **ChangeTMOBJECTTags()**

### **ChangeTMOBJECTTagList()**

Vous pouvez modifier les param'etres d'un objet ToolManager avec cette fonction. L'etat de l'objet sera mis 'a jour pour refl'eter les nouveaux param'etres. A noter: Actuellement, les objets Image ne peuvent ˆetre modifi'es.

### **DeleteTMOBJECT()**

Avec cette fonction, vous pouvez effacer un objet ToolManager. Si l'objet est li'e aux autres objets, ces objets seront avertis pour mettre 'a jour leur 'etats.

### **QuitToolManager()**

Cette fonction dit au gestionnaire de ToolManager de quitter d'es que possible.

La description compl'ete de l'interface de la librairie est disponible au format AutoDoc (voir Paragraphe 7.1 [Docs], page 8).

## 11 Comment d'efinir un raccourci-clavier

Ce chapitre d'ecrit comment d'efinir un raccourci-clavier comme une Chaine de Description d'Entr'ees qui est alors intercept'ee par Commodities. Chaque fois qu'un raccourci clavier est activ'e, Commodities g'en'ere un 'evenement qui est utilis'e par ToolManager pour activer des objets Exec ou pour basculer les objets Dock. Une chaine de description a la syntaxe suivante:

```
[<class>] {[<qualifier>]} [-][upstroke] [<key code>]
```

Tous les mots cl'e sont insensibles aux majuscules.

**class** d'ecrit la classe InputEvent. Ce param'etre est optionnel et si il est manquant, le d'efaut **rawkey** est utilis'e. Voir Paragraphe 11.1 [Classes InputEvent], page 24. Les 'Qualifiers' sont des "signaux" qui peuvent ˆtre positionn'es ou effac'es le temps que le raccourci clavier est activ'e; sinon aucun 'evenement ne sera g'en'er'e. Pour chaque 'qualifier' qui doit ˆtre appel'e, vous fournissez son mot cl'e. Tous les autres 'qualifiers' sont attendus 'a ˆtre effac'es par d'efaut. Si vous voulez ignorer un 'qualifier', mettez juste un - avant son mot cl'e. Voir Paragraphe 11.2 [Qualifiers], page 25.

Normalement, un 'ev'nement d'une 'Hot Key' est g'en'er'e quand une touche est press'ee. Si l'ev'nement doit ˆtre g'en'er'e quand la touche est relach'ee, fournissez le mot cl'e **upstroke**. Quand le fait de presser et de relacher d'une touche doivent g'en'erer un 'ev'nement, utiliser **-upstroke**.

Le code de touche est d'ependant de la classe InputEvent. Voir Paragraphe 11.3 [Codes de touche], page 26.

Note: Choisissez vos touches d'appel *carefully*, parce que Commodities a une priorit'e dans la chaˆme du gestionnaire InputEvent (i.e. sera 'ecras'e par les d'efinitions existantes).

### 11.1 Classes InputEvent

Commodities supporte la plupart des classes InputEvent qui sont g'en'er'es par l'input.device. Cette partie d'ecrit ces classes qui sont les plus pratiques pour les touches d'appel de ToolManager.

**rawkey** C'est la classe par d'efaut et couvre tous les 'ev'nements clavier. Par exemple, **rawkey a** ou **a** cr'ee un 'ev'nement chaque fois que la touche "a" est press'ee. Vous devez

spécifier un code de touche pour cette classe. Voir Paragraphe 11.3.1 [rawkey key codes], page 26.

**rawmouse** Cette classe d'ecrit tous les 'ev'énements des boutons de souris. Vous devez spécifier un code de touche pour cette classe. Voir Paragraphe 11.3.2 [rawmouse key codes], page 27.

**diskinserted**

Les 'ev'énements de cette classe sont g'en'er'es lorsqu'un disque est ins'er'e dans un lecteur. Cette classe n'a pas de codes de touche.

**diskremoved**

Les 'ev'énements de cette classe sont g'en'er'es lorsqu'un disque est sorti d'un lecteur. Cette classe n'a pas de codes de touche.

## 11.2 Qualificateurs

Certains synonymes de mot-cl'es ont 'et'e rajout'es 'a la version 38 de "Commodities". Ceci sont marqu'es avec une \*.

**lshift, left\_shift \***

Touche Shift gauche.

**rshift, right\_shift \***

Touche Shift droite.

**shift** Touche Shift.

**capslock, caps\_lock \***

Touche Majuscule.

**caps** Touche Shift ou touche Majuscule.

**control, ctrl \***

Touche Controle.

**lalt, left\_alt \***

Touche Alt gauche.

**ralt, right\_alt \***

Touche Alt droite.

**alt** 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 *`

Ce mot-clé *doit* être utilisé pour n'importe quelle touche du clavier numérique.

`leftbutton`, `lbutton *`, `left_button *`

Bouton gauche de la souris. Voir la note ci-dessous.

`midbutton`, `mbutton *`, `middlebutton *`, `middle_button *`

Bouton du milieu de la souris. Voir la note ci-dessous.

`rbutton`, `rightbutton *`, `right_button *`

Bouton droit de la souris. Voir la note ci-dessous.

`repeat` Ce qualificateur doit être précisé quand la répétition-clavier est active. Il n'est utile que pour la classe `rawkey` d'événement d'entrée.

Note: La version 37 de "Commodities" contient une erreur qui empêche l'utilisation de `leftbutton`, `midbutton` et `rbutton` comme qualificateurs. Cette erreur a été corrigée dans la version 38.

## 11.3 Codes de touches

Chaque classe d'événement d'entrée a ses propres codes de touche:

### 11.3.1 Key codes for InputEvent class rawkey

Certains mots-clés et des synonymes ont été rajoutés à la version 38 de "Commodities". Ceci sont marqués avec une `*`.

`a-z`, `0-9`, ...

Caractères ASCII.

`f1`, `f2`, ..., `f10`, `f11 *`, `f12 *`

Touches de fonctions.

`up`, `cursor_up *`, `down`, `cursor_down *`

`left`, `cursor_left *`, `right`, `cursor_right *`

Touches fléchées.

`esc`, `escape *`, `backspace`, `del`, `help`

`tab`, `comma`, `return`, `space`, `spacebar *`

Touches spéciales.

`enter`, `insert *`, `delete *`

`page_up *`, `page_down *`, `home *`, `end *`

Touches du clavier num'érique. Chacun de ces codes de touche *doit* utilis'e le qualificateur `numericpad` !

de la souris —

### 11.3.2 Key codes for `InputEvent` class `rawmouse`

Ces mots-cl'e ont 'et'e ajout'es 'a la version 38 de "Commodities". Ils ne sont pas disponibles dans la version 37.

`mouse_leftpress`

Appuyer sur le bouton gauche de la souris.

`mouse_middlepress`

Appuyer sur le bouton du milieu de la souris.

`mouse_rightpress`

Appuyer sur le bouton droit de la souris.

Note: Pour utiliser un de ces codes de touche, vous devez 'egalement pr'eciser la qualificateur correspondant, par exemple

```
rawmouse leftbutton mouse_leftpress
```

## 11.4 Exemples de raccourci-clavier

`ralt t` Maintenir enfonc'e la touche Alt de droite et presser "t"

`ralt lalt t`

Maintenir enfonc'e les touches Alt de gauche et de droite et presser "t"

`alt t` Maintenir enfonc'e la touche Alt et presser "t"

`rcommand f2`

Maintenir enfonc'e la touche Amiga de droite et presser la seconde touche de fonction.

`numericpad enter`

Maintenir enfonc'e la touche Entrer du clavier num'érique.

`rawmouse midbutton leftbutton mouse_leftpress`

Maintenir enfonc'e le bouton du milieu de la souris et presser la bouton gauche de la souris.

**diskinserted**

Insérer une disquette dans n'importe quel lecteur.

## Appendice A Les questions les plus posées sur ToolManager

Voici les questions les plus demandées sur ToolManager:

- Pourquoi ToolManager ne peut-il pas créer de multiples menus ou sous-menus “Outils” ?  
Les menus ou sous-menus multiples ne sont pas actuellement supportés par le système d’exploitation. ■  
Pour les créer, vous devez le **bricoler**, ce qui rend un système instable. Je n’ai pas voulu produire un logiciel instable, alors, je n’implémenterais pas une pareille chose dans ToolManager.
- Les programmes WB ne veulent pas se lancer, mais tous les autres types Exec marchent bien. ToolManager se base sur le programme **L:WBStart-Handler** pour lancer les programmes WB. Il y a deux raisons, pour lesquelles ToolManager ne peut pas exécuter ce programme:  
Le fichier ‘**L:WBStart-Handler**’ n’existe pas. SVP, copier-le depuis l’archive de distribution.  
Le flag d’exécution (e) n’est pas mis sur le fichier. Utilisez la commande suivante pour mettre ce flag : **protect L:WBStart-Handler +e**
- Comment puis-je créer une dock horizontale ?  
Mettez juste le nombre de colonnes aux nombres d’entrées de l’objet dock.
- Comment puis-je créer une fenêtre de sortie pour les programmes CLI ?  
Les fenêtres de sortie peuvent être créées en utilisant le device **CON:**. Utilisez le nom de fichier suivant pour créer une fenêtre auto-ouvrante avec un gadget de fermeture qui ne se ferme qu’après la fin du programme :  
**CON:10/10/640/100/Output-Window/AUTO/CLOSE/WAIT**  
Le device **CON:** a beaucoup d’options, svp consulter votre manuel AmigaDOS pour plus d’informations.
- Comment puis-je mettre les arguments au milieu de la ligne de commande CLI/Arexx ?  
Normalement, tous les arguments sont passés à la ligne de commande. Pour insérer les arguments n’importe où dans la ligne de commande, ToolManager utilise la même syntaxe [], que celle qui est utilisée par la commande AmigaShell **alias**. Aussi, par exemple :  
**Dir [] all**  
insérera tous les arguments avant le mot clé **all**.
- Comment puis-je effacer le lien d’un objet complexe à un objet simple ?  
Après avoir pressé le bouton “xxx Objet”, presser le bouton “Ok” *sans* sélectionner un objet. Cela signifie que vous avez choisi aucun objet, et alors le lien sera effacé.
- Comme puis-je créer des sous-docks ?  
Vous devez utiliser des objets Exec du type Dock. Mettez de pareils objets dans les entrées de votre dock principale et ils pourront ouvrir/fermer les autres docks.

- ToolManager est mort après le lancement d'une commande Réseau.

Il y a actuellement un problème avec le logiciel via réseau, qui n'annule pas les requêtes locales. Aussi, si votre machine est appelée **Hôte1** et vous avez un objet Exec du type Network avec la commande **Objet@Hôte1**, ToolManager bouclera quand vous l'activerez. SVP, utiliser seulement les noms des machines distantes !!!



## Appendice B L'Histoire de ToolManager

### 2.1, Release date 16.05.1993

- New Exec object types: Dock, Hot Key, Network
- New Dock object flags: Backdrop, Sticky
- New object type: Access
- Network support
- Editor main window is now an AppWindow
- Gadget keyboard shortcuts in the preferences editor
- New tooltypes for the preferences editor
- Several bug fixes
- Enhanced documentation

### 2.0, Release date 26.09.1992

- Complete new concept (object oriented)
- (Almost) Complete rewrite
- ToolManager is now split up into two parts
- Main handler is now embedded into a shared library
- Configuration is now handled by a Preferences program
- Configuration file format has changed again :-) It is an IFF File now and resides in ENV:
- Multiple Docks and multi-column Docks
- Docks with new window design
- Dock automatically detects largest image size
- Sound support
- Direct ARexx support for Exec objects
- ToolManager can be used without the Workbench. If the Workbench isn't running, it won't use any App\* features.
- Locale support
- Path from Workbench will be used for CLI tools
- Separate Handler Task for starting WB processes

### 1.0 upto 1.5

Please look into the english documentation for further information.

## Appendice C L'auteur voudrait remercier...

ToolManager est pass'ée par plusieurs 'etapes majeures dans son 'evolution depuis sa premi'ere implementation 'a la mi-1990. Ce d'evveloppement n'aurait pas être possible si je n'avais pas reçu autant de r'eaactions de la part de beaucoup d'utilisateurs de ToolManager. Beaucoup d'id'ees et de caract'eristiques en sont inspir'ees...

Par cons'equent, je voudrais remercier:

Pour les pr'e-tests, les id'ees et le rapport d'erreurs:

La section Amiga de notre club informatique local (Computerclub an der RWTH Aachen), Olaf 'Olsen' Barthel, Georg Hessmann (Gucky), Markus Illenseer (ill), Klaus Melchior, Rickard Olsson (Richie), Matthias Scheler (Tron), Ralph Schmidt (laire), Roger Westerlund (Budda), Juergen Weinelt, Brian Wright (SteveVai), Petra Zeidler (stargazer) et beaucoup d'autres...

Matthew Dillon

Sans votre **excellent** DICE et beaucoup d'autres de vos outils, ToolManager n'existerait pas!

Pour leur excellent travail graphique:

Andreas Harrenberg, Georg Hessmann, Michael "Mick" Hohmann, Markus Illenseer, Oliver Koenen, Klaus Melchior, Rickard Olsson, Jan Peter, Matthias Scheler, Brian Wright

Pour les traductions :

Tomi Blinnikka (suomi), Jorn Halonen (norsk), Dr. Peter Kittel (deutsch), Jasper Kehlet (dansk), Klaus Melchior (eifel), Rickard Olsson (svenska), Rullier Pascal & Marc Schaefer (français), Tor Rune Skoglund (norsk), Reinhard Spisser (italiano), Andrea Suatoni (italiano)

Tous les copains de West Chester:

Pour le d'evveloppement de l'Amiga et de son superbe syst'eme d'exploitation.

Tous les utilisateurs qui m'ont envoy'e de l'argent:

Je n'en avais pas demand'e dans les versions 1.x, mais c'est agr'eable de voir que quelqu'un apprecie mon travail.

Tous les utilisateurs qui m'ont envoy'e un mot:

J'aime vraiment lire vos lettres!

et tous ce que j'ai oubli'e de mentionner...

# Index

## A

Access objets .....	18
Adresse .....	2
Adresse InterNet .....	2
Adresse postale .....	2
AmigaGuide .....	8
ARexx scripts .....	11
Arguments CLI .....	22

## C

Catalogue des fichiers .....	10
Commentaires .....	2
Compilation support .....	11
Concepts .....	5
Concepts des programmes .....	5
Configuration .....	19
Crédits .....	32

## D

Diskinserted .....	24
Diskremoved .....	24
Distribution des fichiers .....	8
Dock objets .....	16
Documentaion des librairies .....	8
Documentation .....	8
Documentation ASCII .....	8
Documentation imprim'ee .....	8
Donations .....	2

## E

E-Mail .....	2
Exec objets .....	12
Exemple .....	6
Exemple d'images .....	9
Exemples de raccourci-clavier .....	27

## G

GetPubName .....	8
GiftWare .....	1
Guide .....	6

## H

Historique .....	31
------------------	----

## I

Icon objets .....	15
Image objets .....	13
Images de Contribution .....	9
InputEvent classes .....	24
Installation (rapide) .....	3
Installation rapide .....	3
Interface librairie .....	23
Interface librairie partag'ee .....	23
Introduction 'a ToolManager .....	4
Introduction aux objets de ToolManager .....	5
Introduction aux raccourci-claviers .....	24

## J

Joueur de Son .....	8
---------------------	---

## K

Key codes for <b>rawkey</b> .....	26
Key codes for <b>rawmouse</b> .....	27

## L

L r'epertoire .....	10
L'interface de la librairie partag'ee de ToolManager .....	23
Langage (fichiers) .....	10
Langages .....	10
Libs r'epertoire .....	10
List: <b>rawkey</b> key codes .....	26
List: <b>rawmouse</b> key codes .....	27
Liste des erreurs .....	2
Liste: Qualificateurs .....	25
Locale r'epertoire .....	10
Localisation .....	10

## M

Menu objets .....	14
-------------------	----

## N

Notes importantes .....	1
-------------------------	---

**O**

Objets .....	12
Objets ToolManager .....	12

**P**

Préférences 'éditeur (de).....	19
Prefs répertoire .....	10
Programmers répertoire.....	11

**Q**

Qualificateurs .....	25
Questions .....	29

**R**

Référence: Distribution des fichiers .....	8
Référence: Interface librairie .....	23
Référence: Objets ToolManager .....	12
Référence: Préférences 'éditeur (de) .....	19
Référence: Raccourci-Clavier .....	24
Répertoire de Documentations.....	8
Répertoire Goodies .....	8
Répertoire Graphics .....	9
Réponses .....	29
Raccourci Clavier .....	24
Rapide installation .....	3
Rawkey .....	24
Rawmouse .....	24
Remerciements .....	32

**S**

Scripts répertoire.....	11
Shell scripts .....	11
Sound objets .....	14
Source code .....	11
Source répertoire.....	11

**T**

T <sub>E</sub> X .....	8
Texinfo .....	8
Tour guid'e.....	6
Traducteurs .....	11
Traductions .....	10
Types d'outils .....	21

**U**

UPD .....	8
-----------	---

**V**

V38 (et plus) caractéristiques.....	1
Versions .....	31
Versions de programmes .....	31

**W**

WBStart 1.2 .....	10
WBStart-Handler .....	10
WBStartup répertoire .....	11

## Table des matieres

<b>1</b>	<b>Important.....</b>	<b>1</b>
<b>2</b>	<b>O‘u envoyer la liste des erreurs, commentaires &amp; donations2■</b>	
<b>3</b>	<b>Comment installer ToolManager 2.1 rapidement...3</b>	
<b>4</b>	<b>Qu’est ToolManager ?.....4</b>	
<b>5</b>	<b>Les concepts de ToolManager.....5</b>	
<b>6</b>	<b>Un tour guid’e ‘a travers ToolManager.....6</b>	
<b>7</b>	<b>Description de tous les fichiers de la distribution..8</b>	
7.1	Le r’epertoire de documentations.....	8
7.2	Le r’epertoire de Goodies.....	8
7.3	Le r’epertoire Graphics.....	9
7.4	Le r’epertoire L.....	10
7.5	Le r’epertoire Libs.....	10
7.6	Le r’epertoire Locale.....	10
7.7	Le r’epertoire Prefs.....	10
7.8	Le r’epertoire Programmers.....	11
7.9	Le r’epertoire Scripts.....	11
7.10	Le r’epertoire Source.....	11
7.11	Le r’epertore WBStartup.....	11
<b>8</b>	<b>R’ef’erence des objets ToolManager.....</b>	<b>12</b>
8.1	Exec objets.....	12
8.2	Objets Image.....	13
8.3	Sound objets.....	14
8.4	Menu objets.....	14
8.5	Icon objets.....	15
8.6	Dock objets.....	16
8.7	Access objets.....	18
<b>9</b>	<b>L’’editeur de pr’ef’erences de ToolManager.....</b>	<b>19</b>
9.1	Fenêtre principale Gadgets.....	19

9.2	Menus fenêtre princiale.....	20
9.3	Fenêtres de modification .....	20
9.4	Fenêtre de cr'eatation d'objets.....	21
9.5	Types d'outils .....	21
9.6	Arguments CLI.....	22
<b>10</b>	<b>L'interface de la librairie partag'ee de ToolManager</b>	<b>23</b>
<b>11</b>	<b>Comment d'efinir un raccourci-clavier.....</b>	<b>24</b>
11.1	Classes InputEvent .....	24
11.2	Qualificateurs.....	25
11.3	Codes de touches.....	26
11.3.1	Key codes for InputEvent class <code>rawkey</code> .....	26
11.3.2	Key codes for InputEvent class <code>rawmouse</code> .....	27
11.4	Exemples de raccourci-clavier.....	27
<b>Appendice A</b>	<b>Les questions les plus pos'ees sur ToolManager</b>	<b>29</b>
<b>Appendice B</b>	<b>L'Histoire de ToolManager .....</b>	<b>31</b>
<b>Appendice C</b>	<b>L'auteur voudrait remercier.....</b>	<b>32</b>
<b>Index.....</b>		<b>33</b>