

Alls Guides

NasGûl

Copyright © CopyrightÂ©1994 NasGûl, Inc.

COLLABORATORS

	<i>TITLE :</i> Alls Guides		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	NasGül	April 16, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Alls Guides	1
1.1	All Guides Documentation	1
1.2	cdps documentation	2
1.3	3dview documentation	2
1.4	description 3dview	3
1.5	equipements 3dview	4
1.6	installation 3dview	4
1.7	fonctionnement 3dview	4
1.8	clickrun documentation	6
1.9	description clickrun	6
1.10	equipements clickrun	6
1.11	installation clickrun	7
1.12	unnamed.1	7
1.13	fonctionnement clickrun	7
1.14	dirw documentation	7
1.15	utils documentation	7
1.16	description utils	7
1.17	equipements utils	8
1.18	installation utils	8
1.19	fonctionnement utils	8
1.20	gui_cliconv documentation	10
1.21	description gui_cc	10
1.22	equipements gui_cc	11
1.23	installation gui_cc	11
1.24	fonctionnement gui_cc	11
1.25	jpeg&ilbm24 documentation	13
1.26	description jpeg&ilbm24	13
1.27	equipements jpeg&ilbm24	14
1.28	installation jpeg&ilbm24	14
1.29	fonctionnement jpeg&ilbm24	14

1.30 mappic documentation	15
1.31 description mappic	16
1.32 equipements mappic	16
1.33 installation mappic	16
1.34 fonctionnement mappic	16
1.35 memfree documentation	17
1.36 description memfree	17
1.37 equipements memfree	17
1.38 installation memfree	17
1.39 fonctionnement memfree	17
1.40 nasscreen documentation	19
1.41 description nasscreen	19
1.42 equipements nasscreen	19
1.43 installation nasscreen	19
1.44 fonctionnement nasscreen	19
1.45 recall documentation	20
1.46 selects documentation	20
1.47 smodule documentation	20
1.48 tm_autodock documentation	21
1.49 description tm_autodock	21
1.50 equipements tm_autodock	21
1.51 installation tm_autodock	22
1.52 fonctionnement tm_autodock	22
1.53 vtext documentation	22
1.54 viewilbm documentation	22
1.55 description viewilbm	23
1.56 equipements viewilbm	23
1.57 installation viewilbm	23
1.58 fonctionnement viewilbm	23
1.59 whatview documentation	23
1.60 description whatview	24
1.61 equipements whatview	24
1.62 installation whatview	24
1.63 fonctionnement whatview	24
1.64 update whatview	26

Chapter 1

Alls Guides

1.1 All Guides Documentation

Sorry, all docs are in french. (i speak/write a very badly ↔ english).

AUX ASSOCIATIONS:

ces programmes sont `_*FREEWARE*_`, c'est a dire qu'ils peuvent être distribués librement par n'importe qui. La seule restriction est que le prix demandé ne dépasse pas le prix du support et de l'envoi.

Il est strictement INTERDIT de faire de l'argent avec ces programmes.

AUX PARTICULIERS:

Le particulier se doit DONNER ces programmes.

Je ne serais aucunement responsable des dégats que peuvent faire ces programmes.

CDPS

3DView

ClickRun

DirW

EUtils

Gui_CliConv

Jpeg&Ilbm24

MappIc

MemFree

NasScreen

Recall

SelectS

Smodule

TM_AutoDock

VText

ViewIlbm

WhatView

Merci a tous les programmeurs qui fournissent leurs sources (←
Pardon aux oubliés).

Nico François pour l'incontournable ReqTools.library (et tous le reste).

Sylvain Rougier et Pierre Carrette pour la WhatIs.library .

Stefan Becker pour ToolManager et la toolManager.library.

Jaba Develoment pour GadToolsBox.

Opal Technology pour l'Opal.library.

Christian A. Weber pour l'iff.library.

Dietmar Eilert pour GoldED.

Barry Wills pour Epp.

Wouter Van Oortmerssen (alias \$#%!) pour son Amiga_E .

Commodore pour son Amiga.

Contact :

NasGûl

Capus André.

74 Rue jules Guesde.

92240 MALAKOFF.

FRANCE.

Tél:46.44.35.28

1.2 cdps documentation

Change l'écran public par défaut. Si aucun paramètre n'est donné ce sera de nouveau le WorkBench par défaut.

1.3 3dview documentation

Description

Equipements

Installation

Fonctionnement

1.4 description 3dview

3Dview est un "viewer" d'objets en 3 dimensions.

Les format d'objets reconnus sont les suivants:

- Cyber Studio v1.0 (ATARI)

plusieurs objets dans le même fichier possible.
Toutes les faces générés par Cyber Studio sont dans le bon sens.

- Cyber Studio v2.0 (ATARI)

plusieurs objets dans le même fichier possible.
Toutes les faces générés par Cyber Studio sont dans le bon sens.

- Imagine (AMIGA)

plusieurs objets dans le même fichier impossible, la base de données
n'aura qu'un seul objet.

- 3Dpro v1.10 Final (AMIGA)

les fichiers objets générés par 3Dpro ne contiennent qu'un objet (pas de ↔
restrictions).
Toutes les faces générés par 3Dpro sont dans le bon sens.
Toutes les faces ayant plus de 3 points sont converties en plusieurs faces à 3 ↔
points.

- Vertex < v1.62a et Vertex > v1.73.1f (AMIGA)

plusieurs objets dans le même fichier impossible, la base de données
n'aura qu'un seul objet.

- Sculpt v2.09 (AMIGA)

plusieurs objets dans le même fichier impossible, la base de données
n'aura qu'un seul objet.

MILLE EXCUSES au possesseurs de LightWave (n'as pas trouvé le monsieur....).

3Dview peut aussi sauver les objets sélectionnés dans les formats suivants:

- VideoScape (AMIGA)

- DXF (PC Beurk...)

Les objets au format .DXF sont construit avec la fonction 3DFACE d'AUTOCAD, ↔
il y a peut
être un autre moyen car comme lenteur on ne fait pas mieux, un petit exemple ↔
juste pour rire.

PC 486+Copro DX 50 Mhz 4 Mo Ram+AutoCad v11+Objet 5000 points 8000 faces = 20 ↔
minutes de rafraichissement.

ATARI STE 68000 8 Mhz 2 Mo Ram+Cyber v2.0+Objet 5000 points 8000 faces = entre 2 ↔
à 4 minutes.

1.5 equipements 3dview

OS 2.XX.
ReqTools.Library v38. ou +

3Dview a été crée et testé sur:

A4000 68EC30+68882 KS 39.106 WB 39.29

1.6 installation 3dview

Copier la reqtools.library dans le dossier libs:.
Copier 3Dview ou vous voulez.

1.7 fonctionnement 3dview

EN CLI:

> 3Dview <fichier> DID <displayid> BLACK.

avec:

<fichier> - Le fichier source.
DID <displayid> - Le format de l'écran de 3Dview.

avec:

<displayid> - BR/BRE/HR/HRE/SHR/SHRE.

BR = 320*256
BRE = 320*512
HR = 640*256
HRE = 640*512
SHR = 1280*256
SHRE = 1280*512

BLACK - Si ce paramètre est présent l'écran sera noir, le dessin blanc.

L'écran (quelque soit son type) a 4 couleurs (depth=2).

EN WB:

ToolTypes

DID=idem cli (BR/BRE/HR/HRE/SHR/SHRE).

PALETTE=BLACK - Si ce tooltype est présent l'écran sera noir, le dessin ↔ blanc.

Une fois les objets chargés les touches suivantes sont actives:

HELP - Montre un résumé des commandes (ce texte en plus court).

Touches du pavé numérique (bonjour a tous les 600...):

NUMPAD 1 - Vue en XOY.

Les objets sont montrés en perspective orthographique suivant le plan XOY.

NUMPAD 2 - Vue en XOZ.

Les objets sont montrés en perspective orthographique suivant le plan XOZ.

NUMPAD 3 - Vue en YOZ.

Les objets sont montrés en perspective orthographique suivant le plan YOZ.

NUMPAD 4 - Inverse les coordonnées en X.

permet d'avoir les objets dans le bon sens. Attention, une permutation des X (5 ↔ devient -5)

n'est pas visible en vue YOZ (logique).

NUMPAD 5 - Inverse les coordonnées en Y.

même remarque mais pour la vue XOZ.

NUMPAD 6 - Inverse les coordonnées en Z.

même remarque mais pour la vue XOY.

NUMPAD + - Zoom In (par pas de 0.01).

La valeur de zoom au départ est de 0.1

NUMPAD + - Zoom Out (par pas de 0.01).

NUMPAD / - Divise le zoom par 2.

NUMPAD * - Multiplie le zoom par 2.

Flèches du curseur:

-
- * HAUT - Rotation positive de toute la base suivant l'axe horizontal (↔ suivant le plan de vue).
 - * BAS - Rotation négative de toute la base suivant l'axe horizontal (↔ suivant le plan de vue).
 - * DROITE - Rotation négative de toute la base suivant l'axe verticale (↔ suivant le plan de vue).
 - * GAUCHE - Rotation positive de toute la base suivant l'axe verticale (↔ suivant le plan de vue).

touches clavier:

-
- * C - Centre la base d'objets en <0,0,0>.
 - L - Charge un nouvel fichier_objet (la base de données est effacée).
 - A - Ajoute un fichier_objet.
 - M - Charge plusieurs fichier_objets. Cette commande permet de charger un ↔ scène de 3Dpro.
 - S - Stop le dessin.
 - I - Donne le nombres d'objets, le total des points et des faces.

Touches de fonctions:

F8 - Sauve les objets sélectionnées au format DXF (PC).

- F9 - Sauve les objets selectionnées au format GEO (VidéoScape).
- F10 - Ouvre la fenêtre d'informations sur les objets.
- ESC - Quitter.

Fenêtre d'informations sur les objets (Uniquement en HR et +):

Cette fenêtre affiche une liste de tous les objets présent dans la base en ↔
donnant:

- le nombres de points de l'objet.
- le nombres de faces de l'objet.
- l'adresse de départ des données des points.
- l'adresse de départ des données des faces.
- MinX,MaxX,MinY,MaxY,MinZ,MaxZ.
- Centre de l'objet (X,Y,Z).
- Type d'objet (Cyber,Imagine...).

2 autre fonctions sont possible:

- Selected - l'objet est selectionné (en bleu).
- Bounded - l'objet n'est pas dessiné lors d'un rafraichissement mais sa place est marqué par un boite blanche l'encadrant.

Conversion en ASCII:

Les commandes marquées d'un * changent les coordonnées des points.

1.8 clickrun documentation

Description

Equipements

Installation

Fonctionnement

1.9 description clickrun

ClickRun permet de lancer un commande a partir d'une liste.

1.10 equipements clickrun

OS 2.XX

ClickRun a été crée et testé sur:

A4000 68EC30+68882 KS 39.106 WB 39.29

1.11 installation clickrun

Ou vous voulez.

1.12 unnamed.1

ClickRun permet de lancer un programme d'un simple click, le programme est lancé avec la commande `exec Execute()`

1.13 fonctionnement clickrun

le fonctionnement est intuitif, lancez le programme est cliquez.

1.14 dirw documentation

Description : DirW est une commande Dir faisant intervenir la WhatIs.library.
Equipements : OS 2.XX+WhatIs.library. (copiez DirW ou vous voulez).
Fonctionnement : DirW <dossier> - Dir Normal avec le type de fichier.
DirW <fichier> COMP <type> - Retourne -1 si le type de fichier est le même sinon 0.

1.15 outils documentation

Description

Equipements

Installation

Fonctionnement

1.16 description outils

EUtils est un aide a la programmation en E. Il fait intervenir les commandes cli suivantes:

- EC (compilateur).
- EPP (pré-processeur).

- Iconvert (Convertisseur d'include assembleur).
- Pragma2Module (Convertisseur de pragma C en binaire).

il permet de compiler, utiliser EPP avec n'importe quel editeur de texte.

1.17 equipements utiles

```
OS 2.XX.
ReqTools.library.
RexxSysLib.library.
```

1.18 installation utiles

Copiez la librairie ReqTools.library dans le dossier libs:.
mettez EUtils ou vous voulez.

1.19 fonctionnement utiles

En cli uniquement :

```
.1>EUtils source.e
```

Eutils ouvre le fichier source et cherche les mots suivants,
ils doivent être évidemment dans des commentaires.

```
ED      "<vôtre editeur>"
EC      "<EC avec options>"
PREPRO  "<EPP avec ces options>"
SOURCE  "<fichier source>" /* doit être le même que le fichier réel */
EPPDEST "<fichier de destination pour EPP>"
EXEC    "<nom de l'executable>"
ISOURCE "<nom de l'include ASM>"
HSOURCE "<nom de l'include C>"
ERROREC "<commande a executer en cas d'erreur avec EC>" /* non fonctionnel */
ERROREPP "<commande a executer en cas d'erreur avec EPP>" /* non fonctionnel ←
*/
VERSION "<numéro de version>"
REVISION "<numéro de révision>"
NAMEPRG  "<nom du programme>"
NAMEAUTHOR "<nom de l'auteur>"
```

voici le bloc que j'utilise :

```
/*
* << AUTO HEADER XDME >>
*/
ED
EC
PREPRO
```

```

SOURCE
EPPDEST
EXEC
ISOURCE
HSOURCE
ERROREC
ERROREPP
VERSION
REVISION
NAMEPRG
NAMEAUTHOR

```

```

*****
* HISTORY :
*****/

```

Un fois tous les paramètres initialisés EUtils lance votre editeur en mode run, ouvre un port Arexx avec les fonctions suivantes:

```

EC      - compile le SOURCE.
EPP     - EPP le SOURCE en EPPDEST est compile EPPDEST, si tout c'est bien
passé Eutils renomme EPPDEST (exe) en EXEC.
INFO    - Requirer avec tous les paramètres.
OPTEPP  - Change les options de EPP.
OPTEC   - Change les options de EC.
ICONV   - Converti ISOURCE.
HCONV   - Converti HSOURCE.
CHI     - Change (par filerequirer) ISOURCE.
CHH     - Change (par filerequirer) HSOURCE.
CHE     - Change (par filerequirer) SOURCE.
CHEPPDEST - Change (par filerequirer) EPPDEST.
NEWSHELL - Commande newshell.
QUIT    - Quit EUtils.

```

A chaque compilation reussi EUtils crée un fichier nommé <source>.header ce fichier contient en binaire les informations suivantes:

```

(---) OBJECT prgheader
( 0)  version:INT          /* version */
( 2)  revision:INT         /* revision */
( 4)  strdate:substructure /* date (JJ-MM-AA) de la dernière compilation ←
*/
(12)  strtime:substructure /* heure (HH:MM:SS) de la dernière ←
compilation */
(20)  auteur:substructure  /* nom de l'auteur */
(40)  nomprg:substructure  /* nom du programme */
(60)  vertag:substructure  /* version string $VER: ..... */

```

un module (Mheader.m) est un pmodule (PMheader.e) sont fourni pour pouvoir lire le fichier <source>.header.

ce qui donnera pour le fichier exemple.e :

```

/*****/
/* bla bla      */
/*****/

```

OPT ...

```

ENUM ..
MODULE 'Mheader'
PMODULE 'Pmodules:PMHeader'

PROC main()
p_DoReadHeader({prg_banner})
ENDPROC
prg_banner:
INCBIN 'exemple.header'

```

une seule variable est globale dans le procédure p_DoReadHeader() :

```

DEF title_req[80]:STRING /* StringF(title_req,'\s v\d.\d @ \s',prg_name, ←
prg_version,prg_revision,prg_author) */

```

mais vous pouvez en sortir d'autre (la date par exemple)

ceci vous permet de ne pas avoir a changer tout vos requesters, alertes ou autres.

1.20 gui_cliconv documentation

Description

Equipements

Installation

Fonctionnement

1.21 description gui_cc

Gui_CLiConv est un "batch_processor" pour convertir des images ↔

Le nombre de format supportés et en rapport avec vos commandes CLI de conversion (Cjpeg/Djpeg/Wasp/Convert etc....). La multisélection est possible.

Etant fait au départ pour convertir des images, les points d'extention sont changés automatiquement.

Les commandes lancées par le gadget "Convert" sont des scripts AmigaDos.

Ex:

```

Commande :JPEG2TGA
Dir_In   :hdl:images/jpeg
Dir_Out  :hdl:images/tga
Pat_In   :#?.jpeg
Pat_Out  :#?.tga

```

Convertiras "hdl:images/jpeg/Ionic5.jpeg" En "hdl:images/tga/ionic5.tga".

Le format des scripts est particulier (cf.
Fonctionnement Gui_CC
) ,
ainsi que leurs localisation (cf.
Installation Gui_CC
).

1.22 equipements gui_cc

OS 2.XX.
Reqtools.library v37 ou +. (Version fournie : 38.1042).

Gui_CC a été crée et testé sur:

A4000 68EC30+68882 KS 39.106 WB 39.29

1.23 installation gui_cc

Copier la Reqtools.library dans le dossier libs:.
Copier Gui_CC et Gui_CC.info ou vous voulez.
Assigner Conv_Scripts: <lecteur/dossier contenant vos scripts> .

1.24 fonctionnement gui_cc

EN CLI:

> Gui_CliConv

ou

> Run <nil: >nil: Gui_CliConv

EN WB:

Double-cliquez sur son icône.

Si le dossier Conv_Scripts: n'est pas assigné Gui_CC quitte automatiquement.

Un fois lancé, La liste des scripts disponibles est visible dans le gadget de liste. En cliquant sur un script, tout les gadgets deviennent opérationnels.

Toutes édition d'un des gadgets entrainera un changement de la ligne de commande lancer par "Convert" .LE SCRIPT RESTERA INCHANGER.
Le seul moyen de changer un script est le menu Edit.

Le menu Viewer lance votre viewer favori, le requestfile s'ouvre dans le 'Dir_Out' de Gui_CC. De plus le viewer étant lancé en mode CLI vous pouvez mettre des paramètres dans le RequestFile.

Ex:

le viewer étant VT (viewtek):

- selectionnez un image en 640*512
- cliquez Ok.

--> L'image est affiché en 320*256 (Scroll possible).

- selectionnez un image en 640*512
- rajouter Hires Lace après le nom de l'image (comme en cli) cliquez Ok.

--> L'image est affiché en 640*512 (Image entièrement visible).

Bien entendu les arguments possible sont ceux de votre viewer.

IMPORTANT:

Les scripts sont des scripts typiquement AmigaDos, seuls quelques commentaires sont nécessaires. (LES 3 PREMIERES LIGNES SONT OBLIGATOIRE).

Ex:

Nom du fichier script :JPEG2HAM

```
.key source,destin
.bra {
.ket }
djpeg030 -gif "{source}" t:temp0
wasp.030.881 -ham t:temp0 "{destin}"
Delete >nil: t:temp0
;COM=JPEG2HAM
;DIR_IN=hdl:images/jpeg
;DIR_OUT=hdl:images/ham
;PAT_IN=#?.jpeg
;PAT_OUT=#?.ham
```

Gui_CC se sert des lignes en commentaires pour les String_gadgets, donc pour la ligne de commande.

TOUTES les commandes sont sourcées dans le dossier Conv_Scripts:, il est donc nécessaire de faire un fichier script même pour une seule commande.

Ex:

Nom du fichier script :JPEG2GIF

```
.key source,destin
.bra {
.ket }
djpeg030 -gif "{source}" "{destin}"
;COM=JPEG2GIF
;DIR_IN=hdl:images/jpeg
;DIR_OUT=hdl:images/gif
;PAT_IN=#?.jpeg
```

```
;PAT_OUT=#?.gif
```

Une fois votre script sélectionné, cliquez sur "Convert".

un sélecteur de fichiers vous permet une multisélection (Shift-Click).

La longueur totale du slider gadget représente le nombres total de fichiers à convertir, le petit rectangle noir représente le fichier

sur lequel Gui_CC est en train de travailler (ce rectangle se déplace de gauche a droite). Les autres gadgets sont désactivés lors de la conversion.

Le nom du fichier en train d'être converti est donné en-dessous du "slider_gadget".

Ajout de fichier Script:

```
-----
```

Pour ajouter un fichier script, vous pouvez :

- editez un fichier déjà existant.
- faire les modifications nécessaires.
- le sauver sous un autre nom (ATTENTION AUX COMMENTAIRES).
- Faire l'entrée menu NewList.

Votre script apparaitra dans la liste.

Si vous copiez un fichier script dans le dossier Conv_Scripts:

- Faire l'entrée menu NewList.
- entrez son nom dans le string_gadget Commande.
- Editez le.
- faire les modifications nécessaires.
- le sauver.

Même si votre éditeur favori fait sauter le bit de script, ce bit est remis lors d'un NewList sur tous les fichiers scripts.

1.25 jpeg&ilbm24 documentation

Description

Equipements

Installation

Fonctionnement

1.26 description jpeg&ilbm24

Jpeg&Ilbm24 est un convertisseur Jpeg<->Ilbm24, il peut sauver des images Ilbm avec le "chunk" special de la carte Opal.

1.27 equipements jpeg&ilbm24

OS 2.XX.

Opal.library v2.2 ou +. (Version fournie : 2.2).

Jpeg&Ilbm24 a été testé sur:

A4000 68EC30+68882 KS 39.106 WB 39.29

1.28 installation jpeg&ilbm24

Copier l' Opal.library dans le dossier libs:.

Copier Jpeg&Ilbm24 et Jpeg&Ilbm24.info ou vous voulez.

1.29 fonctionnement jpeg&ilbm24

EN CLI:

>Jpeg&Ilbm24

EN WB:

Double-cliquez sur l'icone de Jpeg&Ilbm24.

- Choisissez un sens de conversion

JPEG->ILBM24 - du Jpeg en Ilbm 24 bits.

ILBM24->JPEG - du Ilbm 24 bits en Jpeg.

ILBM->ILBM24 - du Ilbm en Ilbm 24 bits.

- Opal Chunk - Le fichier Ilbm resultant contiendras le chunk opal.

- Fast Format - Option de sauvegarde pour la carte 16 millions de couleurs ←
Opal.

IMPORTANT: Si vous n'avez pas la carte Opal ces deux options ne doivent pas ←
être
selectionnées.

Ces deux options sont désactivées lors des conversions Ilbm24->Jpeg

- Jpeg Corrupt - défini les pertes dans le fichier Jpeg.

une valeur de 100 donneras aucune perte d'informations dans le fichier
Jpeg (mais un taux de compression moindre).

Ex:

Jpeg Corrupt=100

fichier source :874174 octets.

fichier destination:255343 octets.

Jpeg Corrupt=75

fichier source :874174 octets.
fichier destination: 91633 octets.

Lors de la conversion le titre de la fenêtre de Jpeg&Ilbm24 indique le numéro de fichier traité sur le nombres total de fichiers a convertir. Le titre de l'écran indique le nom du fichier en cours de conversion. (Aucun break n'est possible lors d'une conversion).

Le premier selecteur de fichiers vous permet de choisir les fichiers sources. Le deuxieme vous permet de choisir le dossier de destination.

Les points d'extention sont rajoutés automatiquement.

Sur des fichiers Ilbm 24 bits - .Ilbm24
Sur des fichiers Jpeg - .jpeg

Si le fichier source a déjà un point d'extention celui-ci est supprimé et est remplacé par l'extention correspondante.Si le fichier source n'a pas de point d'extention celui-ci est rajouté.

Ex:

Conversion Jpeg->Ilbm24

Fichier source : Image.tagada Fichier de destination : image.Ilbm24
Fichier source : Image Fichier de destination : image.Ilbm24

INFORMATION :

Lors de la conversion Jpeg&Ilbm24 ouvre un écran virtuel 24 bits de la taille de l'image,cette écran est un écran spécial (pour la carte Opal).Il est impossible de le voir sur un machine n'ayant pas le hardware (la mienne fait partie du lot),et de plus,la place que prend cet écran dépend de la taille de l'image en source. Donc a moins d'avoir 16 Mo de ram: il est preferable d'avoir les fichiers a traiter sur disque (HD) est non en ram:.

1.30 mappic documentation

Description

Equipements

Installation

fonctionnement

1.31 description mappic

MAppIc permet d'associer une AppIcon a un programme, qui normalement n'en a pas.

1.32 equipements mappic

MAppIc a été programmé en Amiga_E v2.1 (Fred Fish 810) et ce, sur un 4000/68E030+68882 en OS 3.0. Il devrait tourner sans problème en OS 2.xx.

Les libraries utilisées sont celle du système.

1.33 installation mappic

MAppIc est une commande CLI, si jamais vous décidez de la mettre dans votre Startup-sequence elle doit être placée après la commande LoadWB, sinon vous pouvez le mettre n'importe où sur votre HD.

1.34 fonctionnement mappic

Syntaxe:

```
MAppIc <nom icone sans .info> COM <commande associée> NAME <nom de l'icône du WB> [QUIT|TEST|MULTI]
```

Le nom de l'icône doit être donné sans l'extension .info.

Si plusieurs paramètres sont définis dans la commande associée il faut utilisé des " ".

Si aucun nom est donné L'AppIcon n'en auras pas.

L'option QUIT permet de lancer une seule fois la commande associée.

L'option test permet de tester les coordonnées d'une icône.

L'option MULTI permet a MAppIc de se comporter comme une véritable AppIcon.

Pour Quitter MAppIc il suffit de déplacer n'importe quelle icône sur l'icône a enlever (Sauf en mode MULTI).

ASTUCES:

Pour placer une icône en AppIcon, faire une copie en ram: de l'icône a placer, la "sortir" par le WorkBench, la placer où vous voulez sur votre WorkBench. Ensuite "Figer" cette icône (toujours par le WorkBench), il ne vous reste plus qu'à la tester pour avoir ses coordonnées que vous devez rapporter dans les tooltypes de l'icône (POSX et POSY).

Attention

Le Workbench est sensible à l'emplacement de vos icônes, si il ne peut pas la placer (POX POSY) ou si ces tooltypes sont absent, le WorkBench desideras par lui même sa place. Même sans nom il est impossible de "coller" les icônes verticalement, le WorkBench tient compte de la place prise par le nom.

1.35 memfree documentation

Description
Equipements
Installation
fonctionnement

1.36 description memfree

MemFree v 0.0d Documentation

MemFree affiche la mémoire chip/fast total, le plus grand bloc et ce qui reste.

1.37 equipements memfree

MemFree a été programmé en Amiga_E v2.1 (Fred Fish 810) et ce, sur un 4000/68E030+68882 en OS 3.0. Il devrait tourner sans problème en OS 2.xx.
Les libraries utilisées sont celle dy système a part la ReqTools.Library v 38.810 (CAM 665B).

1.38 installation memfree

Copiez MemFree ou vous voulez et le reqtools.library dans le dossier Libs:

1.39 fonctionnement memfree

Fonctionnement:

En CLI:

```
MemFree DELAY 10 [FRONT|PS <nom_pubscreen>].
```

- DELAY defini le temps de rafraichissement de la fenêtre en Tick. (1 tick=1/50 secondes).
- FRONT, la fenêtre de MemFree seas toujours 'ToFront'.
- PS , la fenêtre de MemFree tenteras de s'ouvrir sur l'écran public spécifié.

EN WB:

Une icône Tool avec comme ToolTypes possible:

DELAY=<n> - n en Tick.
 FRONT=[TRUE|FALSE] - Toujours 'ToFront' (Vrai ou faux).
 PS=<nom_pubscreen> - Nom de l'écran public.

MemFree a aussi un port Arexx nommé MemFreePort, les commandes possible sont les suivantes:

FRONT - met MemFree 'ToFront'.
 BACK - met MemFree 'ToBack'.
 YESFRONT - met MemFree toujours 'ToFront'.
 NOFRONT - inverse de YESFRONT.
 FAST - Affiche la mémoire FAST.
 CHIP - Affiche la mémoire CHIP.
 CHANGEPS <nom_pubscreen> - MemFree change d'écran public, si l'écran spécifié n'existe pas MemFree seras sur le WB.
 NEWDELAY <delay> - Définie un nouveau delay (en tick toujours).
 FASTCHIP - Switch entre FAST et CHIP a la même fréquence que delay (un delay de 50 fera inverser CHIP et FAST toutes les secondes).
 ZIP - Fonction ZipWindow.
 QUIT - Quitte MemFree.

Le port AREXX peut être utilisé de la manière suivante:

```
>rx "address 'MemFreePort' FRONT" - Met MemFree 'ToFront'.
```

Les mots entre '' auront les majuscules/minuscules respectées, ceci est très important pour les écrans public (les espaces le seront aussi).

```
>rx "address 'MemFreePort' changeps 'MyPublicScreen'
```

Les commandes AREXX doivent être uniquement en majuscules, pour ce faire il suffit de ne pas les encadrés de ''.

Historique :

v 0.0a - Version Initiale.
 v 0.0b - Ajout de DELAY et PUBSCREEN (en CLI comme en WB).
 v 0.0c - Ajout de FRONT (en CLI comme en WB).
 v 0.0d - Ajout d'un port AREXX.
 Commandes : QUIT/BACK/FRONT/NOFRONT/YESFRONT/FAST/CHIP/
 CHANGEPS <nom_pubscreen>/NEWDELAY <delay>/
 FASTCHIP/ZIP.

v 0.0e - Ajout des menus (GadTools.library) et des requesters de la ReqTools.library.

1.40 nasscreen documentation

Description
 Equipements
 Installation
 Fonctionnement

1.41 description nasscreen

NasScreen ouvre un écran public, qui devient l'écran par défaut. Les menus sont entièrement configurables.

1.42 equipements nasscreen

OS 2.XX

1.43 installation nasscreen

Copiez NasScreen ou vous voulez.

1.44 fonctionnement nasscreen

NasScreen construit ces menus a partir d'un fichier de configuration comme ←
 suis:

```
# Menu For NasScreen
SCREENMODE MODE "SHR" TYPE "PAL" DEPTH "2" DP "1" BP "2" FONT "Tpoaz.font/8"
PALETTE COLO "$787" COL1 "$111" COL2 "$ABB" COL3 "$068"
# SYSTEM UTILITIES
MENU " WorkBench "
ITEM " Utilities "
SUBI " MultiView " STACK "8000" COMM "Sys:Utilities/Multiview"
SUBI " Clock " STACK "4000" COMM "Sys:Utilities/Clock"
SUBI "AmigaGuide " STACK "8000" COMM "Sys:utilities/amigaguide `requestfile`"
```

La ligne SCREENMODE:

```

MODE "BR"      320*256  TYPE "PAL"  DEPTH "number" DP "number"  BP "number" FONT ↔
               "name.font/size"
"BRE"      320*512      "NTSC"                detail pen bock pen
"HR"       640*256
"HRE"      640*512
"SHR"      1280*256
"SHRE"     1280*512

```

La ligne PALETTE COL0 "couleur" COL1 "couleur" COL2 "couleur" COL3 "couleur"

```

MENU "titre du menu"
ITEM "titre" STACK "stack" COMM "commande"
SUBI "titre" STACK "stack" COMM "commande"

```

Toutes les commandes sont lancées avec le fonction SystemTagList() en mode ↔
ASYNCH.

1.45 recall documentation

Description : Recall popup un requester si la date courante a un rapport avec ↔
une date dans

le fichier s:Recall.date.

Equipements : ReqTools.library.

Installation : copiez Recall ou vous voulez, copiez Recall.date dans le dossier ↔
s:

et ajoutez la ligne suivante dans votre user-startup :
Recall

Le fichier recall.date est du type suivant (sans les expicatifs) :

```

23-10-** Anniversaire d'Aubin.      -> tout les 23/10 de chaque année.
**-03-** On est au mois de Mars.    -> tout les mois de mars de chaque année.
01-01-** Bonne Année.              -> tout les 1 janvier.
24-12-** Joyeux Noël.              -> tout les Noël.
11-**-** On est le 11.              -> tout les 11 du mois.
**-**-94 On est en 1994.            -> toute l'année 94.
11-03-94 On est le 11/03/94.       -> le 11-03-94 uniquement.

```

1.46 selects documentation

Description : SelectS permet de selectionné (tofront) un écran (public ou ↔
non).

Equipements : OS 2.XX + reqtools.library.

Installation : Copiez SelectS ou vous voulez.

Fonctionnement : intuitif.

1.47 smodule documentation

Description :

Smodule permet de voir les modules E. Jusqu'a 10 modules sont possible.

Equipements:

OS 2.XX

Foctionnement:

la seule option possible est un fichier de configuration comme suis:

```
emodules:exec/execbase
emodules:intuion/intuition
etc...
```

```
1.>Smodule <fichier de config>
```

1.48 tm_autodock documentation

Description

Equipements

Installation

Fonctionnement

1.49 description tm_autodock

TM_AutoDock Initialise un Dock de ToolManager en parcourant un dossier passé en paramètre. Seuls les fichiers Executables seront mis dans le Dock.

1.50 equipements tm_autodock

OS 2.XX.

ToolManager.Library v3.

WhatIs.Library v3.

TM_AutoDock a été crée et testé sur:

A4000 68EC30+68882 KS 39.106 WB 39.29

1.51 installation tm_autodock

Copier la ToolManager.library et la WhatIs.Library dans le dossier libs:.
Copier TM_AutoDock ou vous voulez.

1.52 fonctionnement tm_autodock

EN CLI:

```
> TM_AutoDock <dossier> COL <nbrs de colonnes> POPUP HOTKEY "alt t" ARG ICON.
```

avec:

```
<dossier>    - Le dossier parcouru.
COL <n>      - Le nombres de colonnes du Dock. (optionnel default n=1).
POPUP       - Le Dock seras ,ou ne seras pas ,enlevé après une sélection.
HOTKEY ou HK <key> - Hotkey d'appel (cf. doc de ToolManager).
ARG         - Le Dock est en AppWindow.
ICON        - Afficheras le Dock avec les icônes au lieu de texte.
              (ATTENTION:Il suffit d'un seul fichier sans icône pour que
              TM_AutoDock quitte).
```

EN WB:

Y'a pas.

Les fichiers contenus dans le dossier sont triés avec la WhatIs.library
seuls les types de fichier suivant sont valides:

```
Exe,Pure Exe,PP40 Exe,PP30 Exe,PP Exe,Script
```

Les fichiers sans icônes seront lancés par ToolManager en mode CLI avec une ↔
sortie.

Les autres (avec icônes) seront lancés en mode WB.

Une fois le Dock affiché vous pouvez le quitter par le programme Exchange, ceci
vous permet de lancer TM_AutoDock avec run.

1.53 vtext documentation

1.54 viewilbm documentation

Description

Equipements

Installation

Fonctionnement

1.55 description viewilbm

ViewIlbm est un viewer ILBM. Il peut sauvez le chunk CAMP dans le fichier ILBM.

1.56 equipements viewilbm

OS 3.XX (visualisation ILBM24 (seuls les 8 premiers bitplanes sont pris en compte, ce qui donne un image en 256 couleurs gris).
iff.library.
reqtools.libray.

1.57 installation viewilbm

Copiez la librairie iff.library dans votre dossier libs:.
mettez ViewIlbm ou vous voulez.

1.58 fonctionnement viewilbm

En CLi uniquement:

ViewIlbm FICHER,HIRES/S,LACE/S,LORES/S,REQ/S,INFO/S,FILEREQ/S

avec :

Fichier - fichier source Ilbm.
HIRES,LACE,LORES - Change le format.
REQ - Choix du format d'écran par screenrequester.
INFO - Information sur le fichier par requester.
FILEREQ - Choix du fichier source par filerequester.

Une fois l'image visible, un click droite quitte ViewIlbm.
si vous voulez sauvez le nouveau CAMP cliquez sur les 2 boutons de la souris (d'abors le gauche puis le droite), l'écran doit flasher, vous pouvez maintenant quittez, le nouveau chunk sera sauvé.

1.59 whatview documentation

Description

Equipements

Installation

fonctionnement

Update Whatview

1.60 description whatview

WhatView vous permet d'un simple clic de la souris de lancer n'importe quelle application en rapport avec le type de fichiers lui étant passé en paramètre.

1.61 equipements whatview

WhatView a été programmé en Amiga_E v2.1 (Fred Fish 810) et ce, sur un 4000/68E030+68882 en OS 3.0 (KS 39.106 WB 39.29). Il devrait tourner sans problème en OS 2.xx. Les libraries utilisées sont celle dy système a part la WhatIs.Library v 3.5 (Fred Fish 843) et la ReqTools.library v 38.1042 (CAM 790B).

1.62 installation whatview

Vous pouvez mettre WhaView ou vous voulez, mais il faut s'avoir que WhatView lit les toolstypes de son icone. Du cli il faut donc lui mettre en paramètre son icone sans le .info.

Ex:

```
1.>Whatview Hd0:WbStartup/Whatview
```

Si l'envie vous prend de le mettre dans le dossier Wbstartup ne pas oublier le tooltype DONOTWAIT.

copier les libraries WhatIs.library et reqtools.library dans le dossier Libs:.

1.63 fonctionnement whatview

WhatView n'utilise que les tooltypes de son icône comme fichier de configuration. Plusieurs tooltypes sont interne :

```
DONOTWAIT      - Si WhatView est dans le dossier Wbstartup
DEFDIR=opt     - Défini le dossier par défaut pour les icônes prédéfi-
                nies (WhatIs.library). Ceci permet de soulager le
                dossier env:sys.
                Si ce tooltypes est absent le dossier par défaut (Env:Sys) sera pris.
POSX=position x - Détermine la position en x de l'Appicon.
POSY=position y - Détermine la position en y de l'Appicon.
```

Shift-cliquez sur les icônes des fichiers puis faire Outils->WhatView (ou ← déplacer les icones sélectionnées sur l'icône de whatview) , la fenêtre de WhatView vous permettant ← de choisir l'action voulue.

Si aucun fichier n'est cliqué la fenêtre de Whatview apparaîtra.

- WHATVIEW, INFO, ADDICON, EXECUTE, QUIT.

Même résultat que la sélection d'un des gadgets de la fenêtre.

quelques exemples:

```
1.>alias WVR rx "address 'WhatViewPort' []"
```

faites WVR WHATVIEW ou INFO ou ADDICON ou EXECUTE ou QUIT - Sélection par ↵
filerequester.

ou:

```
1.>rx wv ram:* info      (script arexx la pattern est celle de la arp.library) ↵  
.
```

1.64 update whatview

La doc que vous venez de lire est valable pour les versions 0.11 et inférieure.

A partir de la version 0.15 Whatview a été réécrit pour profiter du WBStart- ↵
Handler Port. ↵

Whatview se compose donc maintenant de 3 programmes:

- Whatview.
- WVprefs.
- Whatis.

Whatview a toujours besoin de son icone mais stocke désormais ses informations ↵
dans le
fichier Env:Whatview.prefs.

Les ToolTypes possible pour Whatview sont:

- DONOTWAIT -> Si vous mettez Whatview Dans le WBStartup.
- DEFDIR -> Dossier des icones.
- DEFPREFSDIR -> Dossier contenant WVprefs.
- POSX et POSY -> Position de l'icone.
- MAXARG -> Nombres d'arguments maximum (20 args est le MAX).

Les Commandes associées aux types de fichiers sont lancer beaucoup plus
proprement que les versions antérieures.

Les arguments sont triés suivant leurs types est la commande associée n'est
lancer qu'une seule fois contrairement au version précédentes. Vous pouvez
donc mélanger des images, du texte, etc.. sans avoir 20 Background CLI.

ATTENTION le fonction Execute de Whatview est toujours la fonction Exec ↵
Execute().

Les modes de lancement ainsi que les fichier Whatview.prefs, est géré
par le programme WVPrefs.

La liste de gauche donne tous les ID de votre système.

La liste de droite est la liste sauvegardé.

