

M2IARexx

COLLABORATORS

	<i>TITLE :</i> M2IAR _{exx}		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 14, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	M2IARexx	1
1.1	M2I version V / AREXX	1
1.2	AREXX et M2I	2
1.3	Exécuter une commande AREXX depuis M2I	2
1.4	Débuter avec AREXX	3
1.5	Index des commandes	5
1.6	Adresse de M2I	8
1.7	Codes de Retour	8
1.8	La gestion des espaces et des quotes	9
1.9	Les Arguments 'Cherche-Icône'	10
1.10	Instructions de Projet	13
1.11	Récupération d'Informations	14
1.12	Manipulation des Tiroirs / Fenêtres	15
1.13	Boîtes de Messages / Requêtes	16
1.14	Exécution de Commandes / Outils	16
1.15	Edition d'un menu	17
1.16	Manipulation des Ecrans Publics	18
1.17	Instructions Inclassables	18
1.18	OPENDRW	19
1.19	CLOSEDRW	19
1.20	CLOSESUBS	20
1.21	ICONIFY	20
1.22	REMICON	21
1.23	EXECTOOL	21
1.24	INTERNAL	22
1.25	QUIT	22
1.26	WBTASKS	23
1.27	OPENMENU	23
1.28	SAVEMENUAS	23
1.29	SAVEMENU	24

1.30 REQUEST	24
1.31 MESSAGE	25
1.32 ERROR	26
1.33 FLUSH	26
1.34 SOUND	26
1.35 PLAYFILE	27
1.36 POPUP	27
1.37 GETPATH	28
1.38 GETNAME	28
1.39 GET	29
1.40 SET	30
1.41 COUNT	32
1.42 REQDIR	33
1.43 REQFILE	33
1.44 REQSTRING	34
1.45 REQMULTIFILE	34
1.46 OPENPUBDRW	35
1.47 PUBLICDRW	35
1.48 GETPUBLIC	36
1.49 SETDEFPUB	36
1.50 PUBTOFRONT	36
1.51 VERSION	36
1.52 EDMODE	37
1.53 LOCKUSER	37
1.54 UNLOCKUSER	37
1.55 REDRAW	38
1.56 GETCHG	38
1.57 ADDCHG	38
1.58 ADDDRW	39
1.59 ADDTOOL	39
1.60 ADDICON	40
1.61 DRWTOFRONT	40
1.62 GETPUBSIZE	40
1.63 GETMOUSE	41
1.64 GETWINSIZE	41
1.65 MOVEITEM	41
1.66 LASTSELECTPATH	42
1.67 LASTSELECTNAME	43
1.68 DELETE	43
1.69 WBINFO	44
1.70 AUTOSCAN	44

Chapter 1

M2IARexx

1.1 M2I version V / AREXX

=====

M2I version V / AREXX

Copyright (C) 1991-1999 par Thomas PIMMEL

Utilisation de M2I avec AREXX MAJ 5.78

=====

SOMMAIRE

Introduction

AREXX et M2I

Exécuter une commande AREXX depuis M2I

Débuter avec AREXX

Index des commandes
Principes généraux

Adresse de M2I

Codes de Retour

La gestion des espaces et des quotes

Les Arguments 'Cherche-Icône'
Classement par fonction

Instructions de Projet

Récupération d'Informations

Manipulation des Tiroirs / Fenêtres

Boîtes de Messages / Requêtes

Exécution de Commandes / Outils

Edition d'un menu

Manipulation des Ecrans Publics

Instructions Inclassables

1.2 AREXX et M2I

AREXX ET M2I

Soyons clairs : tout ce l'utilisateur peut faire à la souris AREXX peut le faire. Avec tout l'avantage que peut apporter AREXX, c'est à dire l'introduction de commandes qui manipulent M2I à l'intérieur d'un BASIC.

Mais AREXX donne aussi accès à des fonctions de M2I qui ne sont pas accessibles à la souris : c'est le cas de la manipulation des écrans publics.

M2I comporte 51 commandes AREXX.

Exemple de possibilités :

- Ajout automatique d'Outils d'après le contenu d'un répertoire sur votre disque.
- Ouverture d'un Tiroir sur n'importe quel écran public
- Mémoriser des configurations M2I et en changer d'un simple clic.
- Changer les paramètres de nombreux Tiroirs ou Outils en même temps.

etc...

1.3 Exécuter une commande AREXX depuis M2I

EXECUTER UNE COMMANDE AREXX DEPUIS M2I

Rien de plus facile. Editez ou créez un Outil. Changez le gadget cyclique qui indique SHELL / WBENCH / AREXX.

Placez-le sur AREXX.

Dans la zone "Outil" placez le nom de votre script AREXX.

Si votre script est situé en Rexx: ne placez pas de chemin.
Si l'extension du script est ".rexx" ou celle précisée
dans les préférences "DIVERS" ne précisez pas d'extension.

Dans la zone Arguments placez les arguments à envoyer
au script, ou rien.

Si vous voulez une sortie mettez-en une. Si votre sortie
doit être interactive veuillez à y mettre une fenêtre CON:.

Vous pouvez également affecter une touche de fonction
à cette commande dans la zone "Clavier".

Cliquez OK.

Passez en mode utilisation.

Double-Cliquez sur votre Outil.

Si AREXX n'est pas actif M2I va tenter de lancer RexxMast,
comme vous l'avez précisé dans les préférences DIVERS,
zone "RexxMast".

Votre script se déroule.

Si AREXX renvoie une erreur, M2I l'affiche en clair (48 erreurs
possibles).

L'erreur la plus fréquente est "Programme non trouvé".
AREXX n'a pas trouvé votre script.

1.4 Débuter avec AREXX

DEBUTER AVEC AREXX

ARexx n'est rien de plus qu'un programme, qui lit des textes
où l'on va inscrire des commandes, qu'il va exécuter.

La particularité d'ARexx est qu'il intègre une convention,
permettant la communication entre les programmes.

CE QUI EST INDISPENSABLE

Le programme RexxMast, fourni avec votre Amiga. Normalement il
devrait se trouver dans votre répertoire "System", si je n'ai
pas trop forcé sur le houblon.

La library "RexxSysLib" qui se trouve en LIBS:, elle aussi
fournie généreusement avec votre machine.

Un éditeur de texte, le bon vieux Ed faisant l'affaire, ou tout autre éditeur.

CE QUI EST UTILE

Les deux programmes TS et TE permettent respectivement de mettre ou enlever le mode "Pas à Pas" d'AREXX.
Très pratique en cas d'erreur dans un script AREXX.

OU AREXX CHERCHE-T-IL SES SCRIPTS ?

En principe dans une assignation REXX:, mais en donnant un chemin complet à M2I ARexx peut les chercher où vous voulez.

A QUOI RESSEMBLE UN SCRIPT AREXX ?

Un script ARexx doit toujours commencer par un commentaire.
Cette obligation un peu curieuse n'est pas stupide. Voici un exemple de script qui ne fait rien, mais qui marche (!!!)

```
/* Ceci est un commentaire AREXX */
```

MAJUSCULE / MINUSCULE

En règle générale ARexx y est insensible, donc M2I aussi.
A une exception près : l'adresse de la tâche que l'on pilote.
Pour M2i cette adresse est : "M2IPort.Rexx".

PILOTER M2I

Voici le script minimum pour piloter M2I :

```
/* On ne fait encore rien...  
Mais c'est un début! */
```

```
OPTIONS RESULTS /* Autorise M2I à renvoyer des résultats */
```

```
ADDRESS "M2IPort.Rexx" /* Les commandes suivantes d'adresseront à M2I */
```

```
EXIT /* On sort la commande "exit" est optionnelle */
```

UN PREMIER SCRIPT

```
/* Ce script compte le nombre d'icônes dans votre menu  
et affiche un message dans M2I */
```

```
OPTIONS RESULTS  
ADDRESS "M2IPort.Rexx"
```

```
/* Compte le nombre d'icônes */  
Count "*" 
```

```
/* Affiche un message */  
Message "Vous avez" RESULT "icône(s) dans M2I"  
  
EXIT      /* on sort */
```

LES VARIABLES

RC signifie toujours sous AREXX le code de retour, à 0 en cas de succès.
RESULT désigne le retour de la fonction appelée. Dans le cas de
notre exemple, RESULT a été renvoyé par la fonction "Count" de
M2I.

POUR EN SAVOIR PLUS

Vous pensez bien que mon propos n'est pas d'écrire une doc
d'AREXX!

Il en existe en Shareware, très complètes. Maintenant que
vous avez les bases minimales, je pense que vous êtes à
même de pouvoir lire les scripts fournis avec M2I.

Bon courage, et n'hésitez pas à me faire parvenir vos productions!

1.5 Index des commandes

INDEX DES COMMANDES

Dans l'ordre de l'alphabet :

```
ADDCHG  
  
ADDDRW  
    "Label" "Cherche-Icône"  
  
ADDICON  
    "Icône" "Cherche-Icône"  
  
ADDTOOL  
    "Label" "Cherche-Icône"  
  
AUTOSCAN  
    "Cherche-Icône"  
  
CLOSEDRW  
    "Cherche-Icône"  
  
CLOSESUBS  
    "Cherche-Icône"  
  
COUNT  
    "Cherche-Icône"
```

```
DELETE
    "Cherche-Icône"

DRWTOFRONT
    "Cherche-Icône"

EDMODE
    [ON/OFF]

ERROR
    "Message"

EXECTOOL
    "Cherche-Icône"

FLUSH

GET
    MotClé "Cherche-Icône"

GETCHG

GETMOUSE
    X/Y

GETNAME
    "Cherche-Icône"

GETPATH
    "Cherche-Icône"

GETPUBLIC
    "PubScreen"

GETPUBSIZE
    W/H "PubScreen"

GETWINSIZE
    W/H "Cherche-Icône"

ICONIFY
    "Cherche-Icône"

INTERNAL
    "Commande"

LASTSELECTNAME

LASTSELECTPATH

LOCKUSER

MESSAGE
    "Message"

MOVEITEM
```

```
[Méthode] "Cherche-Icône" "Cherche-Icône"

OPENDRW
    "Cherche-Icône"

OPENMENU
    ["Nom du menu"]

OPENPUBDRW
    "PubScreen" "Cherche-icône"

PLAYFILE
    "Fichier"

POPUP

PUBLICDRW
    "Cherche-Icône"

PUBTOFRONT
    ["PubScreen"]

QUIT

REDRAW
    "Cherche-Icône"

REMICON
    "Cherche-Icône"

REQDIR
    ["Répertoire"]

REQFILE
    ["Fichier"]

REQMULTIFILE
    "Répertoire" "Motif" ["Message de fenêtre"]

REQSTRING
    "Message"

REQUEST
    [GADGETS "Gad1|Gad..."] "Message"

SAVEMENU
    ["Nom du menu"]

SAVEMENUAS
    ["Nom du menu"]

SET
    MotClé "Valeur" "Cherche-Icône"

SETDEFPUB
    ["PubScreen"]
```

```
SOUND
    [ON|OFF]

UNLOCKUSER

VERSION

WBINFO
    "Cherche-Icône"

WBTASKS
```

1.6 Adresse de M2I

Adresse AREXX de M2I :

```
"M2IPort.Rexx"
```

(Minuscules / Majuscules à respecter scrupuleusement).

Extension conseillée des scripts M2I :

```
".M2I"
```

1.7 Codes de Retour

CODES DE RETOUR

Codes de retour généralement renvoyés par M2I. Ces codes de retour sont contenus dans la variable RC.

```
30 Erreur de mémoire
20 Commande inconnue
11 Mauvais Arguments
10 Erreur lors de l'exécution de la commande
6 Retour de "Cherche-Icône", aucun élément trouvé
5 Avertissement
1 Oui (retourné par
    REQUEST
)
0 Pas d'erreur / Non (retourné par
    REQUEST
```

)

NOTE : Si RC~=0, la variable RESULT n'est pas initialisée.

1.8 La gestion des espaces et des quotes

LES ESPACES ET LES QUOTES

M2I gère très bien les labels et les "Cherche-icône" qui contiennent des espaces.

Mais encore faut-il qu'il soit informé qu'un espace se trouve au milieu d'une chaîne!

Voici comment procède AREXX avec une chaîne qu'il envoie à un programme :

```
/* Exemple SET MotClé "Valeur" "Cherche-icône" */
```

```
Variable=":xxx/yyy/zzz.t"
```

```
set label "Mon Label" Variable
```

```
Exit
```

Voici ce que M2I reçoit :

```
SET LABEL Mon Label :xxx/yyy/zzz.t
```

Les mots sans quote sont mis en majuscule, s'ils correspondent à une variable AREXX le contenu de cette variable est substitué et les quotes sont simplement effacées. M2I cherchera l'icône Label et ne trouvera rien.

Voici comment on doit procéder :

```
set label "'Mon Label'" Variable
```

M2I recevra

```
SET LABEL 'Mon Label' :xxx/yyy/zz.t
```

Vous pouvez aussi inverser la nature des quotes :

```
set label '"Mon Label"' Variable
```

SECURISER LES VARIABLES

Vous pouvez par prudence, quand vous n'êtes pas maîtres de ce que va contenir une variable, entourer cette variable de quotes :

```

REQSTRING "Comment appeler le premier Tiroir de la racine ?"

IF RC=0 THEN
  DO
    NOM="' 'RESULT'"
    SET LABEL NOM ":1"
  END

M2I recevra, si l'utilisateur à renvoyé "Mon Label"

SET LABEL 'Mon Label' :1

```

1.9 Les Arguments 'Cherche-Icône'

LES ARGUMENTS DE TYPE "Cherche-Icône"

"Cherche-Icône" est un argument qui permet de spécifier une ou un ensemble d'icônes, c'est à dire des Tiroirs ou des Outils.

Sauf exception, une commande acceptant un argument "Cherche-Icône" sera exécutée autant de fois que d'icônes correspondant à "Cherche-Icône".

Syntaxe exacte :

```
Cherche-Icône = ":[ ][[path] [*]/...][label] [*][.1|d|t]"
```

M2I n'est pas sensible aux Majuscules/Minuscules dans une "Cherche-Icône".

Nous prendrons l'exemple de la commande OPENDRW (ouvrir Tiroir) dont la syntaxe est OPENDRW "Cherche-Icône", pour expliquer le fonctionnement de "Cherche-Icône".

1. Chercher une icône depuis la Racine

":" désigne la racine. Pour ouvrir le répertoire Racine vous pouvez donc écrire :

```
OPENDRW ":"
```

Vous pouvez ensuite préciser le label d'un autre Tiroir :

```
OPENDRW ":xxx"
```

Le Tiroir nommé "xxx" de la racine est ouvert.

Vous pouvez également préciser des sous-répertoires :

```
OPENDRW ":xxx/yyy" /* ouverture du Tiroir yyy contenu dans xxx */
```

A l'infini...

```
OPENDRW ":xxx/yyy/zzz"
```

NOTE : Si vous disposez de deux répertoires nommés ":xxx", les deux seront ouverts.

2. Spécifier un type d'icône.

"Cherche-Icône" considère que le caractère "/" suit obligatoirement un Tiroir. En se qui concerne le dernier élément de "Cherche-Icône", sa nature n'est par défaut pas déterminée.

Si vous voulez forcer le type d'icône demandé, faites suivre le dernier élément de "Cherche-Icône" par :

```
".d" l'objet demandé est un Tiroir (drawer)
".t" l'objet demandé est un Outil (tool)
```

```
OPENDRW ":xxx/yyy.d" /* Désigne le Tiroir yyy */
```

Que se passe-t-il lorsqu'on passe en argument un Outil à une commande qui demande un Tiroir ?

Les commandes en règle générale considèrent que vous leur demandez d'opérer sur le Tiroir parent de cet Outil.

```
OPENDRW ":xxx/zzz.t" /* revient à ouvrir le Tiroir xxx */
```

3. Forcer un argument unique

Par défaut la commande est appelée pour chaque correspondance avec "Cherche-Icône" (sauf exception).

En précisant ".1" à la fin de "Cherche-Icône" vous vous assurez d'une exécution unique.

```
OPENDRW ":xxx.1" /* Ouvrira le premier Tiroir de la racine
s'appelant xxx */
```

".1" peut être associé avec une spécification de type :

```
".1d" ou ".dl" signifie premier Tiroir
```

```
".1t" ou ".tl" signifie premier Outil
```

4. Les icônes sans label

Vous pouvez indiquer à la place d'un label un numéro d'ordre dans un Tiroir. Exemple :

```
OPENDRW ":3" /* Troisième icône de la racine */
```

ATTENTION : le numéro d'ordre ne dépend pas du type d'icône.

OPENDRW ":3.d" échouera si la troisième icône n'est pas un Tiroir!

Exemple valide : OPENDRW ":xxx/4/yyy/3"

Dans tous les cas la valeur du RC sera 6 si "Cherche-Icône" ne trouve pas ce que vous lui avez demandé.

(Exception : la commande

COUNT

renverra 0 dans RESULT

si "Cherche-Icône" ne trouve rien, ce qui est logique).

5. Les jokers

Rien ne vous oblige à mettre le label complet d'une icône dans "Cherche-Icône". Faites dans ce cas suivre le début du nom par une "*".

OPENDRW ":xx*"

Ouvrira les Tiroirs dont le label commence par "xx"

Vous pouvez aussi mettre juste une étoile :

OPENDRW ":*"

Ouvre tous les Tiroirs de la racine et, si un Outil est présent sur la racine, la racine elle-même.

Les spécifications de type restent valables :

OPENDRW ":x*/yyy/z*.d"

NOTE : Si vous remplacez un label par "*" les icônes sans labels sont concernés également par la commande.

Vous pouvez également remplacer une lettre par un "?".

OPENDRW ":xxx?" désigne aussi bien xxxl, que xxxf...

6. Les omissions de chemins

Vous n'êtes pas obligés de partir de la racine pour donner un argument "Cherche-Icône".

OPENDRW "xxx/*.d" désigne tous les Tiroirs contenus dans un Tiroir appelé xxx.

Si plusieurs Tiroirs s'appellent "xxx", la commande agira pour chacun de ces Tiroirs.

Vous pouvez également ne pas mettre de chemin du tout.

OPENDRW "zzz.t" Ouvrira le Tiroir qui contient l'Outil "zzz"

NOTE : "*" désigne donc l'ensemble des icônes,
"*.t" l'ensemble des Outils,
"*.d" l'ensemble des Tiroirs.

7. Conclusion

Les commandes acceptant un argument "Cherche-Icône" sont donc très puissantes, et permettent beaucoup de fantaisies.

En règle générale un Tiroir suivit d'un nom d'Outil suffit à individualiser un Outil particulier. Par exemple, chez moi trois icônes répondent au nom de "Devpac". Un Tiroir, un script AREXX et un Outil SHELL.

"Devpac.d" est unique, c'est le Tiroir de mes Outils de langage.

"ARexx/Devpac" est unique également.

"Devpac/Devpac" aussi...

NOTE : Lorsque vous utilisez une "*" essayez d'optimiser la recherche.

Exemple :

OPENDRW "*.d" fait la même chose que OPENDRW "*", mais elle n'est pas appelée autant de fois.

Dans le cas de

REDRAW
c'est encore plus

flagrant.

REDRAW "*" va redessiner une fenêtre un fois par Outil qu'elle contient!

REDRAW "*.d" est plus approprié.

1.10 Instructions de Projet

INSTRUCTIONS DE PROJET

Regroupe toutes les instructions ayant un rapport avec le "Menu Projet".

Chargement / Sauvegarde

```
OPENMENU  
  ["Nom du menu"]
```

```
SAVEMENU  
  ["Nom du menu"]
```

```
SAVEMENUAS  
  ["Nom du menu"]
```

Quitter

```
QUIT  
RAPPEL :
```

```
GETNAME  
  ":" renvoie le nom du Menu
```

1.11 Récupération d'Informations

```
RECUPERATION D' INFORMATIONS
```

Regroupe toutes les instructions qui n'agissent pas sur M2i mais ne font que renvoyer des renseignements sur le menu.

```
COUNT  
  "Cherche-Icône"
```

```
GET  
  MotClé "Cherche-Icône"
```

```
GETCHG
```

```
GETMOUSE  
  X/Y
```

```
GETNAME  
  "Cherche-Icône"
```

```
GETPATH  
  "Cherche-Icône"
```

```
GETWINSIZE  
  W/H "Cherche-Icône"
```

```
LASTSELECTNAME
```

```
LASTSELECTPATH
```

VERSION

WBTASKS

RAPPEL :

GETNAME

":" renvoie le nom du Menu

SOUND

sans argument renvoie l'état de
la tâche Son.

EDMODE

sans argument renvoie le mode
actuel de fonctionnement de M2I.

1.12 Manipulation des Tiroirs / Fenêtres

MANIPULATION DES TIROIRS (DRAWERS)

Ouverture / fermeture

OPENDRW

"Cherche-Icône"

DRWTOFRONT

"Cherche-Icône"

CLOSEDRW

"Cherche-Icône"

CLOSESUBS

"Cherche-Icône"

Icônification

ICONIFY

"Cherche-Icône"

REMICON

"Cherche-Icône"

POPOP

AutoDir

AUTOSCAN

VOIR EGALEMENT

```
SET
    XPOS|YPOS "Cherche-Icône"

GETWINSIZE
    W/H "Cherche-Icône"

FLUSH
```

1.13 Boites de Messages / Requêtes

```
BOITES DE MESSAGES / REQUETES
```

Messages

```
ERROR
    "Message"

MESSAGE
    "Message"
```

Requêtes

```
REQUEST
    [GADGETS "Gad1|Gad..."] "Message"

REQSTRING
    "Message"
```

Requêtes de fichier / Répertoire

```
REQDIR
    ["Répertoire"]

REQFILE
    ["Fichier"]

REQMULTIFILE
    "Répertoire" "Motif" ["Message de fenêtre"]
```

1.14 Exécution de Commandes / Outils

```
EXECUTION DE COMMANDES / OUTILS
```

Commandes Internes

```
INTERNAL  
  "Commande"
```

Outils

```
EXECTOOL  
  "Cherche-Icône"
```

1.15 Edition d'un menu

EDITION D'UN MENU

Pour visualiser à l'écran les changements

```
REDRAW  
  "Cherche-Icône"
```

Instructions nécessitant un appel
à la commande REDRAW

```
ADDDRW  
  "Label" "Cherche-Icône"
```

```
ADDICON  
  "Icône" "Cherche-Icône"
```

```
ADDTOOL  
  "Label" "Cherche-Icône"
```

```
DELETE  
  "Cherche-Icône"
```

```
MOVEITEM  
  [Méthode] "Cherche-Icône" "Cherche-Icône"
```

Instruction nécessitant parfois un appel
à la commande REDRAW

```
SET  
  MotClé "Valeur" "Cherche-Icône"
```

Instructions Optionnelles

```
ADDCHG
```

```
EDMODE  
    [ON/OFF]
```

```
LOCKUSER
```

```
UNLOCKUSER
```

1.16 Manipulation des Ecrans Publics

```
MANIPULATION DES ECRANS PUBLICS
```

```
GETPUBLIC  
    "PubScreen"
```

```
GETPUBSIZE  
    W/H "PubScreen"
```

```
OPENPUBDRW  
    "PubScreen" "Cherche-icône"
```

```
PUBLICDRW  
    "Cherche-Icône"
```

```
PUBTOFRONT  
    ["PubScreen"]
```

```
SETDEFPUB  
    ["PubScreen"]
```

VOIR EGALEMENT

L'exécutable WinPatch

1.17 Instructions Inclassables

```
INSTRUCTIONS INCLASSABLES
```

Economiser la mémoire

```
FLUSH  
    Sons
```

```
PLAYFILE  
    "Fichier"
```

```
SOUND  
    [ON|OFF]
```

Divers

```
WBINFO  
  "Cherche-Icône"
```

1.18 OPENDRW

```
OPENDRW "Cherche-Icône"
```

Ouvre le Tiroir spécifié, ou, si un Outil est donné en paramètre le Tiroir qui abrite l'Outil.

```
Retour : 0 succès  
         10 erreur lors de l'ouverture d'une fenêtre  
         (Arrêt immédiat de Cherche-Icône)
```

```
Résultat : Nombre de fois où la procédure a été appelée  
           par "Cherche-Icône"
```

NOTE : Si le Tiroir était icônifié, l'icône est retirée du Workbench.

Voir

```
CLOSEDRW  
  "Cherche-Icône"
```

```
CLOSESUBS  
  "Cherche-Icône"
```

1.19 CLOSEDRW

```
CLOSEDRW "Cherche-Icône"
```

Ferme le Tiroir spécifié, ou, si un Outil est donné en paramètre le Tiroir qui abrite l'Outil.

```
Retour : 0
```

```
Résultat : Nombre de fois où la procédure a été appelée  
           par "Cherche-Icône"
```

NOTE : - Si le Tiroir est déjà fermé, la commande est ignorée.

- Les icônes de la fenêtre ne sont pas libérées de la mémoire. Vous devez utiliser la commande

```
FLUSH
Voir
OPENDRW
    "Cherche-Icône"
```

```
CLOSESUBS
    "Cherche-Icône"
```

1.20 CLOSESUBS

```
CLOSESUBS "Cherche-Icône"
```

Ferme tous les Tiroirs fils du Tiroir indiqué en paramètre.

Retour : 0

Résultat : Nombre de fois où la procédure a été appelée
par "Cherche-Icône"

NOTE : - A la différence de CLOSEDRW, CLOSESUBS libère les icônes
des Tiroirs fermés.

- CLOSESUBS ne ferme que les fils, pas le tiroir donné en paramètre n'est pas fermé.

Voir

```
OPENDRW
    "Cherche-Icône"
```

```
CLOSESUBS
    "Cherche-Icône"
```

1.21 ICONIFY

```
ICONIFY "Cherche-Icône"
```

Icônifie le Tiroir spécifié, ou, si un Outil est donné
en paramètre, le Tiroir qui abrite l'Outil.

Retour : 0 succès
10 Erreur lors de l'icônification

Résultat : Nombre de fois où la procédure a été appelée

par "Cherche-Icône"

NOTE : Si le Tiroir est déjà icônifié, la commande est ignorée.

Voir

```
REMICON
    "Cherche-Icône"
```

1.22 REMICON

```
REMICON "Cherche-Icône"
```

Enlève du Workbench l'icône du Tiroir spécifié, ou, si un Outil est donné en paramètre du Tiroir qui abrite l'Outil.

Retour : 0

Résultat : Nombre de fois où la procédure a été appelée
par "Cherche-Icône"

Voir

```
ICONIFY
    "Cherche-Icône"
```

1.23 EXECTOOL

```
EXECTOOL "Cherche-Icône"
```

Exécute LE TOOL donné en paramètre. N'accepte pas toutes les commandes internes. Fonctionne aussi bien en mode Shell que WBench.

Un seul Outil sera exécuté lors d'une commande.

Retour : 5 L'icône est un Tiroir ou un Outil sans commande
0 Ok
10 Erreur de lancement (ou commande interne interdite)

Résultat : Exécutable lancé

NOTE : - Refuse les commandes faisant appel à :
OPEN, SAVE, PARENT, EDIT, WBINFO
Utilisez les commandes ARexx équivalentes.

- Par sécurité, EXECTOOL n'exécute qu'un seul outil, le premier qui correspond à "Cherche-Icône".

Voir

Instructions de Projet

1.24 INTERNAL

INTERNAL "Commande"

Exécute la commande interne donné en paramètre.

Retour : 5 Abandon de l'utilisateur (Préférences)
0 Commande trouvée et autorisée
10 Commande inconnue

Résultat : #####

Commandes Internes :

ABOUT
FLUSH
ICONIFY
MAIN
MISCPREFS
NEW
PENPREFS
PREFS
QUIT
SPY
SNDBREAK
SNDPREFS

(Voir M2I.guide pour en savoir plus)

NOTE : Refuse les commandes :
OPEN, SAVE, PARENT, EDIT, WBINFO
Utilisez les commandes ARexx équivalentes.

Voir

Instructions de Projet

1.25 QUIT

QUIT

Quitter le programme (équivalent du menu "quitter")

Retour : 0

Résultat : Nombre de tâches WBench en attente
(peut être nul)

1.26 WBTASKS

WBTASKS

Retourne le nombre de tâches WBench en cours.

Retour : 0

Résultat : Nombre de tâches WBench en attente
(peut être nul)

1.27 OPENMENU

OPENMENU ["Nom du menu"]

Ouvre le menu spécifié, ou ouvre une requête de fichier si aucun nom est spécifié.

Retour : 5 Abandon par l'utilisateur
0 Pas d'erreur
10 erreur lors de l'ouverture du menu

Résultat : Nom du Menu si retour = 0

Voir

SAVEMENU
["Nom du menu"]

SAVEMENUAS
["Nom du menu"]

1.28 SAVEMENUAS

```
SAVEMENUAS ["Nom du menu"]
```

Ouvre une requête de fichier, si un nom est spécifié il sera proposé à l'utilisateur.

```
Retour : 5 Abandon par l'utilisateur
         0 Pas d'erreur
         10 erreur lors de la sauvegarde du menu
```

```
Résultat : Nom du Menu si retour = 0
```

Voir

```
OPENMENU
  ["Nom du menu"]
```

```
SAVEMENU
  ["Nom du menu"]
```

1.29 SAVEMENU

```
SAVEMENU ["Nom du menu"]
```

Sauve avec le nom par défaut (celui du chargement en principe). Si "nom de menu" est précisé, le menu est sauvé sous le nouveau nom.

```
Retour : 5 Pas de nom par défaut!
         0 Pas d'erreur
         10 erreur lors de la sauvegarde du menu
```

Lors d'un prochain "Save" sans argument, le nom par défaut est celui que vous venez de préciser.

```
Résultat : Nom du Menu si retour = 0
```

Voir

```
OPENMENU
  ["Nom du menu"]
```

```
SAVEMENUAS
  ["Nom du menu"]
```

1.30 REQUEST

```
REQUEST [GADGETS "Gad1|Gad..."] "Message"
```

Ouvre une requête avec le contenu de "Message" et

par défaut deux gadgets (Oui/Non)

```
Retour : 1 Oui
         0 Non
         11 Erreur dans les arguments
```

```
Résultat : #####
```

NOTE : Usage de la commande étendue.

Si vous spécifiez le mot clé "GADGETS" en début des arguments M2I s'attend à trouver les gadgets qui doivent composer le choix de l'utilisateur, séparés par le caractère "|". Le code de retour est alors renvoyé en fonction du nombre de gadgets :

```
Premier Gadget : 1
```

```
Deuxième Gadget : 2
```

```
xième Gadget : x
```

```
Dernier Gadget : 0
```

Attention : Pas plus de 9 gadgets, pour éviter la confusion avec un retour d'erreur.

Exemple :

```
REQUEST GADGETS "Manger|Dormir|Programmer" "Ce soir, vous voulez :"
```

```
1 = "Manger"
2 = "Dormir"
0 = "Programmer"
```

GESTION DES ESPACES

Si vous utilisez des espaces dans les gadgets, pensez à doubler vos quotes ;

Exemples :

```
REQUEST GADGETS "'Choix 1|Choix 2'" "Que voulez vous ?"
```

ou

```
REQUEST GADGETS '"Choix 1|choix 2"' "Que voulez-vous ?"
```

1.31 MESSAGE

```
MESSAGE "Message"
```

Ouvre une requête MESSAGE avec le contenu de "Message" et le gadget OK

```
Retour : 1 Ok
         (10 Erreur!) pas encore en place
```

```
Résultat : #####
```

Voir

```
ERROR
      "Message"
```

1.32 ERROR

```
ERROR "Message"
```

Ouvre une requête ERREUR avec le contenu de "Message" et le gadget OK

```
Retour : 1 Ok
         (10 Erreur!) pas encore en place
```

```
Résultat : #####
```

Voir

```
MESSAGE
      "Message"
```

1.33 FLUSH

FLUSH

Purge la mémoire sans information de l'utilisateur.

```
Retour : #####
```

```
Résultat : Quantité de mémoire libérée en octets
```

1.34 SOUND

```
SOUND [ON|OFF]
```

Installe ou supprime la tâche son, ou renvoie l'état de cette tâche.

Si un argument existe :

```
Retour : 5 L'état ne change pas
```

```
0 Ok
10 erreur, demande impossible
```

Si pas d'argument :

```
1 Son en place
0 Pas de son
```

Résultat : #####

Voir

```
PLAYFILE
  "Fichier"
```

1.35 PLAYFILE

```
PLAYFILE "Fichier"
```

Joue le son dont le nom est "Fichier"

```
Retour : 5 Le son est inactif
0 Ordre envoyé à la tâche son
```

Résultat : #####

Voir

```
SOUND
  [ON|OFF]
```

1.36 POPUP

```
POPUP
```

Dé-Icônifie M2I après une commande "IconifyAll".

```
Retour : 5 M2I n'était pas icônifié
0 Ordre lancé
```

Résultat : #####

Voir

```
INTERNAL
  "Commande"
Avec commande = "Iconify"
```

1.37 GETPATH

```
GETPATH "Cherche-Icône"
```

Retourne le chemin complet d'une icône.

```
Retour : 5 Pas de chemin (répertoire racine)
         0 Succès
         10 Erreur (mémoire)
```

Résultat : Chemin depuis la racine de l'icône

NOTE : Getpath ne retourne que le chemin de la première icône qui correspond à "Cherche-Icône"

Voir

```
GETNAME
    "Cherche-Icône"

LASTSELECTPATH

LASTSELECTNAME
```

1.38 GETNAME

```
GETNAME "Cherche-Icône"
```

Retourne le nom d'une icône suivi de l'extension .T ou .D
Si "cherche-Icône" désigne la racine (":") retourne le nom du menu.

```
Retour : 0
```

Résultat : Nom de l'icône (ou numéro si pas de nom)

NOTE : GetName ne retourne que le nom de la première icône qui correspond à "Cherche-Icône"

Voir

```
GETPATH
    "Cherche-Icône"

LASTSELECTNAME

LASTSELECTPATH
```

1.39 GET

GET MotClé "Cherche-Icône"

Avec MotClé

Tiroir ou Outil :

TYPE type (0/1) (Tiroir/Outil)
LABEL label de l'icône
ICON nom de l'icône

Tiroir :

XPOS position X de fenêtre
YPOS position Y de fenêtre
LOCK fenêtre figée (0/2/4/6) (voir note)
LMOD mode de chargement des icônes (0/1/2)
IPLACE place des icônes (0/1/2) (Haut / Bas / Centrées)
WLIST contenu des fenêtres (0/1/2/3/4/5)
(0, icônes horizontales, 1 verticale, 2 ou 3 textes,
4 boutons horizontaux, 5 boutons verticaux)
LC nombre de lignes / colonnes (1-9)
WINDOW fenêtre ouverte (0/1) (Non/Oui)
APPICON icône dans wbench (0/1) (Non/Oui)
ITEM sous-items (0/1) (Non/Oui)
DRAWERTYPE type de tiroir

DRAWERTYPE=0 Tiroir Simple, DRAWERTYPE<>0 AutoDir

Valeurs à additionner :

0/1 Type (Simple / AutoDir)
0/2 WB (Automatique/Force WBench)
0/4 Tri (Aucun / alphabet)
0/8 Tri (2) (Aucun / sur type)
0/16/32 Fichiers (icône/tous/ignore)
64 Force
128 Répertoire déjà examiné

Exemple : DRAWERTYPE=21 (1+4+16)

DIRECTORY répertoire à examiner (AutoDir)
FILTER filtre (autodir)
DEFTOOL outil par défaut (autodir)

Outil :

TOOL outil
ARG arguments
PATH chemin par défaut de l'Outil

IO sortie
STACK taille de la pile
PRI priorité (-128 à +127)
WB tâche wbench
HOTKEY touche d'appel
SOUND fichier son

Retour : 0 Succès
5 Chaîne vide
10 Mauvais type d'icône

Résultat : D'après le type

NOTE : Cherche-Icône n'exécute la commande que pour une seule icône.

NOTE : Fenêtres figées (lock)

0 signifie normale
2 signifie sans barre de titre
4 sera sans bord (mais apparence normale)
6 (2+4) Sans bord ni barre de fenêtre, look 3D

Exemples :

```
GET STACK "dev*.T"  
GET ICON ":3/PIPO"
```

Voir

```
SET  
MotClé "Valeur" "Cherche-Icône"
```

1.40 SET

```
SET MotClé "Valeur" "Cherche-Icône"
```

Avec MotClé

Tiroir / Outil :

LABEL label de l'icône
ICON nom de l'icône

Tiroir :

XPOS position X de fenêtre
 YPOS position Y de fenêtre
 (la position à l'écran change immédiatement si
 la fenêtre est ouverte. YPOS uniquement)
 LOCK fenêtre figée (0/2/4/6) voir note 4
 LMOD mode de chargement des icônes (0/1/2)
 IPLACE place des icônes (0/1/2) (haut/bas/centrée)
 WLIST contenu des fenêtres (0/1/2/3/4/5)
 (0, icônes horizontales, 1 verticales, 2 ou 3 textes,
 4 boutons horizontaux, 5 boutons verticaux)
 LC nombre de lignes / colonnes (1-9)
 DRAWERTYPE type de tiroir

DRAWERTYPE=0 Tiroir Simple, DRAWERTYPE<>0 AutoDir

Valeurs possibles :

0/1 Type (Simple / AutoDir)
 0/2 WB (Automatique/Force WBench)
 0/4 Tri (Aucun / alphabet)
 0/8 Tri (2) (Aucun / sur type)
 0/16/32 Fichiers (icône/tous/ignore)
 64 Force
 128 Répertoire déjà examiné (mis par M2I)

L'addition des différents paramètres donne la valeur
 de DRAWERTYPE. Toute valeur paire EST INTERDITE,
 à par zéro!

DIRECTORY répertoire à examiner (AutoDir)
 FILTER filtre (autodir)
 DEFTOOL outil par défaut (autodir)

Outil :

TOOL Outil
 ARG arguments
 PATH chemin par défaut de l'Outil
 IO sortie
 STACK taille de la pile (0 à)
 PRI priorité (-128 à +127)
 WB tâche wbench (0 SHELL /1 WB /2 AREXX)
 HOTKEY touche d'appel
 FLG Flags (0 / 1) voir note 1
 SOUND fichier son

Retour : 0 Succès
 10 Mauvais type d'icône

Résultat : Nombre d'icônes traitées

NOTE 1:

Pour l'instant M2I ne connaît qu'un seul Flag, qui permet de rendre la touche d'appel "résidente" pour un outil.

Exemples :

```
SET ICON "" "*" Toutes les icônes sont par défaut.
```

```
SET WB "1" "Dev*.T" Les Outils correspondants à "Dev*.T" doivent se lancer en mode WBench
```

NOTE 2:

LABEL et ICON sont ignorés dans le cas de la racine (Exemple : SET LABEL "Pipo" ":")

NOTE 3: Valeur avec des espaces ou des quotes.

```
SET LABEL ' "Mon Label"' ":1" (Mon Label)
SET LABEL "'Mon Label'" ":1" (Mon Label)
SET LABEL ' "L''oiseau"' ":1" (L'oiseau)
SET LABEL "'L''oiseau'" ":1" (L'oiseau)
```

etc...

NOTE 4: Fenêtres figées (lock)

0 signifie normale

2 signifie sans barre de titre

4 sera sans bord (mais apparence normale)

6 (2+4) Sans bord ni barre de fenêtre, look 3D

Voir

```
GET
MotClé "Cherche-Icône"
(pour les valeurs)
```

1.41 COUNT

```
COUNT "Cherche-Icône"
```

Retourne le nombre d'icônes qui correspondent à "Cherche-Icône"

Retour : 0 Succès

Résultat : Nombre d'icônes.

1.42 REQDIR

```
REQDIR ["Répertoire"]
```

Retourne un répertoire choisi par l'utilisateur.

```
Retour : 0 Succès  
        5 Abandon
```

```
Résultat : Répertoire (avec chemin complet)
```

Voir

```
REQFILE  
    ["Fichier"]
```

```
REQMULTIFILE  
    "Répertoire" "Motif" ["Message de fenêtre"]
```

```
REQSTRING  
    "Message"
```

1.43 REQFILE

```
REQFILE ["Fichier"]
```

Retourne un fichier choisi par l'utilisateur.

```
Retour : 0 Succès  
        5 Abandon
```

```
Résultat : Fichier (avec chemin complet)
```

Voir

```
REQDIR  
    ["Répertoire"]
```

```
REQMULTIFILE  
    "Répertoire" "Motif" ["Message de fenêtre"]
```

```
REQSTRING  
    "Message"
```

1.44 REQSTRING

```
REQSTRING "Message"
```

Retourne un texte rentré par l'utilisateur.

```
Retour : 0 Succès  
        5 Abandon
```

```
Résultat : Chaîne (max. 200 caractères,  
            terminateur non-compris)
```

Voir

```
REQDIR  
      ["Répertoire"]
```

```
REQFILE  
      ["Fichier"]
```

```
REQMULTIFILE  
      "Répertoire" "Motif" ["Message de fenêtre"]
```

1.45 REQMULTIFILE

```
REQMULTIFILE "Répertoire" "Motif" ["Message de fenêtre"]
```

Renvoie une chaîne de nom de fichiers choisis par l'utilisateur.

```
Retour : 0 Succès  
        5 Abandon  
       10 Erreur (mauvais répertoire)  
       30 Mémoire
```

```
Résultat : Chaîne
```

NOTE : La chaîne est composée du répertoire choisi suivi des noms de fichier séparés par un espace. Le répertoire est toujours indiqué, terminé par ":" ou "/". Le répertoire renvoyé n'est pas forcément le même que le répertoire indiqué dans l'argument de la commande.

Exemples de retours :

```
"WBench:PIPO/ UnFichier"
```

```
"Ram: UnFichier DeuxFichier TroisFichier"
```

Voir

```
REQDIR  
      ["Répertoire"]
```

```
REQFILE  
  ["Fichier"]
```

```
REQSTRING  
  "Message"
```

1.46 OPENPUBDRW

```
OPENPUBDRW "PubScreen" "Cherche-icône"
```

Ouvre un Tiroir dans l'écran public précisé dans l'argument.

```
Retour : 0 succès  
         5 Tiroir déjà ouvert ou écran public non-trouvé  
         10 erreur lors de l'ouverture d'une fenêtre  
         (Arrêt immédiat de Cherche-Icône)
```

```
Résultat : Nombre de fois où la procédure a été appelée  
           par "Cherche-Icône"
```

```
Exemple : OPENPUBDRW "WorkBench" ":"
```

NOTE : - Fonctionnement identique à OPENDRW

- N'ouvre que les tiroirs encore fermés

Voir

```
OPENDRW  
  "Cherche-Icône"
```

```
CLOSEDRW  
  "Cherche-Icône"
```

```
CLOSESUBS  
  "Cherche-Icône"
```

1.47 PUBLICDRW

```
PUBLICDRW "Cherche-Icône"
```

Retourne le nom de l'écran public dans lequel la fenêtre est ouverte

```
Retour : 0 Succès  
         5 écran par défaut
```

```
Résultat : Nom de l'écran public
```

1.48 GETPUBLIC

```
GETPUBLIC "PubScreen"
```

Tester existence d'un écran public

```
Retour : 0 Il existe  
        5 Non-Trouvé
```

1.49 SETDEFPUB

```
SETDEFPUB ["PubScreen"]
```

Définir l'écran par défaut du système (sans argument l'écran par défaut de M2I devient l'écran par défaut du système)

```
Retour : 0
```

Attention :

Pas de retour d'erreur!

Utilisez GETPUBLIC si vous voulez être sûr que cet écran existe.

Voir

```
GETPUBLIC  
"PubScreen"
```

1.50 PUBTOFRONT

```
PUBTOFRONT ["PubScreen"]
```

Mettre l'écran public en avant-plan. Si argument nul, l'écran de M2I par défaut est en avant plan (5.35!).

```
Retour : 0 Succès  
        5 Non-trouvé
```

Voir

```
GETPUBLIC  
"PubScreen"
```

1.51 VERSION

```
VERSION
```

Retourne la version de M2I

Retour : 0

Résultat : Version de la forme 5.xx

1.52 EDMODE

EDMODE [ON/OFF]

Passer en mode édition (ON) ou en mode Exécution (OFF)

Retour : 0 Succès
5 Le menu était déjà dans ce mode

Sans argument renvoie le mode :

Retour : 0

Résultat : 0 Edition
1 Utilisation

1.53 LOCKUSER

LOCKUSER

Bloque les actions de l'utilisateur en vue d'une édition du menu par AREXX.

Retour : 0

Résultat : #####

Voir

UNLOCKUSER

1.54 UNLOCKUSER

UNLOCKUSER

Débloque les actions de l'utilisateur après une édition du menu par AREXX.

Retour : 0 Succès
5 N'était pas bloqué

Résultat : #####

Voir

LOCKUSER

1.55 REDRAW

REDRAW "Cherche-Icône"

Redessine le Tiroir spécifié, ou, si un Outil est donné en paramètre le Tiroir qui abrite l'Outil.

Ne tient compte que des Tiroirs (fenêtres) ouverts.

Retour : 0 succès

Résultat : Nombre de fois où la procédure a été appelée
par "Cherche-Icône"

1.56 GETCHG

GETCHG

Renvoie le nombre de changements effectués depuis la dernière sauvegarde.

Retour : 0 succès

Résultat : Nombre de changements

Voir

ADDCHG

1.57 ADDCHG

ADDCHG

Ajoute 1 au compteur de changements.

Retour : 0 succès

Résultat : Nombre de changements

Voir

GETCHG

1.58 ADDDRW

```
ADDDRW "Nom" "Cherche-Icône"
```

Ajouter 1 Tiroir (et un seul) du nom "Nom" dans le chemin "Cherche-Icône"

Si "Cherche-Icône" est un Tiroir insert en queue

Si "Cherche-Icône" est un Outil, insert après cet Outil.

```
Retour : 0 succès  
        30 pas assez de mémoire
```

```
Résultat : ##### (pour l'instant)
```

Voir

```
ADDICON  
        "Icône" "Cherche-Icône"
```

```
ADDTOOL  
        "Label" "Cherche-Icône"
```

1.59 ADDTOOL

```
ADDTOOL "Nom" "Cherche-Icône"
```

Ajouter 1 Outil (et un seul) du nom "Nom" dans le chemin "Cherche-Icône"

Si "Cherche-Icône" est un Tiroir insert en queue.

Si "Cherche-Icône" est un Outil, insert après cet Outil.

```
Retour : 0 succès  
        30 pas assez de mémoire
```

```
Résultat : ##### (pour l'instant)
```

NOTE : Stack par défaut fixée à 4000, pas d'Outil.

Voir

```
ADDICON  
        "Icône" "Cherche-Icône"
```

```
ADDDRW  
        "Label" "Cherche-Icône"
```

1.60 ADDICON

```
ADDICON "Icône" "Cherche-Icône"
```

Ajouter 1 Outil ou un Tiroir (et un seul) en se basant sur l'icône "Icône" dans le chemin "Cherche-Icône"

Si "Cherche-Icône" est un Tiroir insert en queue.

Si "Cherche-Icône" est un Outil, insert après cet Outil.

```
Retour : 0 succès  
        10 Erreur
```

```
Résultat : Label de l'icône
```

NOTE : Routine fonctionnant exactement comme le "drag and drop" dans une fenêtre.

Voir

```
ADDDRW  
    "Label" "Cherche-Icône"
```

```
ADDTOOL  
    "Label" "Cherche-Icône"
```

1.61 DRWTOFRONT

```
DRWTOFRONT "Cherche-Icône"
```

Mettre le Tiroir en avant-plan.

Si "Cherche-Icône" est un Outil mettre le Tiroir parent en avant-plan.

```
Retour : 0 succès
```

```
Résultat : Nombre de fenêtres en avant-plan  
          (peut être égal à 0)
```

NOTE : Ignore les Tiroirs fermés

1.62 GETPUBSIZE

```
GETPUBSIZE W/H "PubScreen"
```

Renvoie la taille en Largeur (W) ou Hauteur (H) de l'écran public concerné.

Exemple : GETPUBSIZE H "Workbench"

Retour : 0 succès
5 Ecran introuvable

Résultat : Taille en pixel

1.63 GETMOUSE

GETMOUSE X/Y

Renvoie la position en X ou Y de la souris dans l'écran actif.

Retour : 0 succès
5 Pas d'écran actif

Résultat : Position à l'écran

1.64 GETWINSIZE

GETWINSIZE W/H "Cherche-Icône"

Renvoie la taille en Largeur (W) ou Hauteur (H) d'une fenêtre ouverte indiquée dans "Cherche-Icône".

Retour : 0 succès
5 Fenêtre fermée

Résultat : Taille en pixel

Exemple : GETWINSIZE W ":PIPO/BIMBO.D"

1.65 MOVEITEM

MOVEITEM [Méthode] "Cherche-Icône" "Cherche-Icône"

=> MOVEITEM [BEFORE/AFTER/LAST] "SOURCE" "DESTINATION"

Déplace les items spécifiés dans source vers la destination.

Retour : 0

Résultat : Nombre d'icônes réellement déplacées ou 0

Avec Méthode :

Sans spécification de méthode la copie se fait dans le Tiroir spécifié en destination, en tête, ou, si la destination est un Outil, en tête du Tiroir père de cet Outil.

Si LAST, copie identique à la précédente, mais en queue
(Attention : Le résultat renvoyé par MOVEITEM n'est pas exploitable si la copie se déroule d'une fenêtre vers le même fenêtre. En effet une même Icône peut être déplacée plusieurs fois)

Si AFTER, copie après l'icône spécifiée dans destination

Si BEFORE, copie avant l'icône spécifiée dans destination

NOTE : Une tentative de copie d'un père dans un fils est ignorée.
La source peut être multiple, la destination est forcément unique.

Exemple (1) :

```
MOVEITEM AFTER "*.t" ":"
```

Exemple (2) :

```
/* Rotation des icônes d'une fenêtre */  
MOVEITEM LAST "Pipo/*" "Pipo.d"
```

1.66 LASTSELECTPATH

LASTSELECTPATH

Renvoie le chemin de la dernière icône sélectionnée par l'utilisateur.

Retour : 0 succès
5 Pas de sélection
30 Erreur de mémoire

Résultat : Chemin de la dernière icône sélectionnée

Voir

LASTSELECTNAME

1.67 LASTSELECTNAME

LASTSELECTNAME

Renvoie le nom de la dernière icône sélectionnée par l'utilisateur.

Retour : 0 succès
5 Pas de sélection
30 Erreur de mémoire

Résultat : Nom de la dernière icône sélectionnée

Exemple de script :

```
/* utilisation conjointe de LASTSELECTPATH et LASTSELECTNAME */
```

```
ADDRESS "M2IPort.Rexx"  
OPTIONS RESULTS
```

```
LOCKUSER
```

```
RESULT=""  
NOM=""
```

```
LASTSELECTPATH  
IF RESULT~="" THEN NOM=RESULT"/"
```

```
LASTSELECTNAME  
NOM=NOM""RESULT
```

```
IF NOM="" THEN ERROR "Pas de sélection"  
ELSE MESSAGE "Sélection : "NOM
```

```
UNLOCKUSER
```

```
EXIT
```

Voir

LASTSELECTPATH

1.68 DELETE

```
DELETE "Cherche-Icône"
```

Détruit l'icône précisé dans "Cherche-icône"

Si "Cherche-Icône" est un Tiroir, détruit toute l'arborescence de ce Tiroir.

Retour : 0 succès

Résultat : Nombre d'icônes détruites

NOTE : Si un Tiroir est détruit, le résultat ne tient pas compte de toutes les icônes réellement éliminées.

1.69 WBINFO

WBINFO "Cherche-Icône"

Appelle la fonction "Information" du Workbench pour chaque icône d'Outil rencontrée.

Retour : 0 fonction appelée
10 Tiroir donné en paramètre

Résultat : Nombre de fois où la procédure a été appelée
par "Cherche-Icône"

NOTE : Fonctionne exactement comme la commande interne "WBINFO" ou le point "WB Info" du menu "Icônes".

1.70 AUTOSCAN

AUTOSCAN "Cherche-Icône"

Examine le répertoire indiqué dans la zone répertoire d'un Tiroir "AutoDir", ou, si un Outil est donné en paramètre le Tiroir qui abrite l'Outil.

Retour : 0 succès

Résultat : Nombre d'Outils ou Tiroirs créés, ou 0

NOTE : Si un tiroir abrite déjà des icônes ou si la fenêtre est ouverte cette commande sera sans effet.

Voir

OPENDRW
"Cherche-Icône"
