

ARexxSupervisor

COLLABORATORS

	<i>TITLE :</i> ARexxSupervisor		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 17, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ARexxSupervisor	1
1.1	Documentation d'ARexxSupervisor: Table des Matières	1
1.2	Cioyright,distribution et garanties	1
1.3	Crédits et remerciements	2
1.4	Configuration nécessaire	2
1.5	Qu'est-ce qu'ARexxSupervisor	3
1.6	Lancer ARexxSupervisor	3
1.7	Lancer ARexxSupervisor à partir du CLI	3
1.8	Lancer ARexxSupervisor à partir du Workbench	4
1.9	La fenêtre d'ARexxSupervisor	4
1.10	Gadget ? / Touche ? : Information	5
1.11	Gadget Aide / touche Help: Affiche le fichier d'aide	5
1.12	Gadget I : Iconification	5
1.13	Gadget Q / touche Q / Gadget de fermeture: Quitter le programme	6
1.14	Gadget Halte / touche H: Stoppe l'exécution de tous les programmes ARexx	6
1.15	Gadget Trace globale / touche G: force le mode trace pour tous	6
1.16	Gadget Port / touche P: nom du port de destination	7
1.17	Gadget Type / touche Y: type du REXXMsg	7
1.18	Envoie une commande	7
1.19	Appelle une fonction	8
1.20	Ferme la console de traçage	8
1.21	Ouvre la console de traçage	8
1.22	Ajoute une entrée au clipboard	8
1.23	Enlever une entrée du clipboard	9
1.24	Ajoute une bibliothèque	9
1.25	Retire une bibliothèque	9
1.26	Gadget FileExt / touche X: Extension de fichier	9
1.27	Gadget StdIn / touche é: Canal d'entrée standard	9
1.28	Gadget StdOut / touche O: Canal de sortie standard	10
1.29	Gadgets ARGx / touches 0..F: Chaînes arguments	10

1.30	Gadget NoIO / touche N: modifier flag NoIO	10
1.31	Gadget Result / touche R: modifier flag Result	10
1.32	Gadget String / touche S: modifier flag String	10
1.33	Gadget Token / touche T: modifier flag Token	11
1.34	Gadget Send message / touche M: Envoie le message à l'hôte	11
1.35	Gadget Result1: Code de retour principal	11
1.36	Gadget Result2: Code de retour secondaire.	11
1.37	Traduction	12
1.38	Améliorations futures	12
1.39	Contacteur l'auteur	12

Chapter 1

ARexxSupervisor

1.1 Documentation d'ARexxSupervisor: Table des Matières

```
                Bienvenue à ARexxSupervisor 1.10, par Frédéric DELACROIX. ↔
                Ceci est le
fichier de documenation qui doit toujours accompagner le programme lors
d'une diffusion publique. Et maintenant, comme tout fichier guide qui se
respecte, voici une TDM:
```

```
TABLE DES MATIERES
```

```
COPYRIGHT, DISTRIBUTION et GARANTIES
    <-- C'est très important

CREDITS et REMERCIEMENTS

CONFIGURATION NECESSAIRE
    <-- Je vous conseille d'y jeter un oeil

QU'EST-CE QU'AREXXSUPERVISOR ?

LANCER AREXXSUPERVISOR

LA FENETRE D'AREXXSUPERVISOR

AMELIORATIONS FUTURES

TRADUCTION

CONTACTER L'AUTEUR
    <-- N'oubliez pas !
```

1.2 Cioyright, distribution et garanties

ARexxSupervisor 1.10 a été entièrement écrit par F.DELACROIX sur le merveilleux Devpac 3 d'Hisoft. JE RESERVE TOUS LES DROITS. Cependant, le programme est distribué en SHAREWARE. Cela signifie que la distribution

est autorisée (et encouragée) tant que:

- AUCUN profit ne soit réalisé au-delà de frais de copie&envoi MINIMAUX (vous ne pouvez pas demander plus que Fred Fish pour un seul Amigalib Disk)

- L'archive en ENTIER (ou, disons, tous les fichiers) soit (soient) distribuée (és) SANS AUCUNE MODIFICATION. Il y a toutefois une exception: Comme ARexxSupervisor est distribué en deux langues (pour l'instant), le français et l'anglais, vous pouvez distribuer que l'une d'elles. Cependant, je préférerais que vous distribuiez les deux si vous diffusez ARexxSupervisor sur une bibliothèque DP internationale très répandue.

N'oubliez pas que ARexxSupervisor est SHAREWARE, ce qui signifie que si vous continuez à l'utiliser après une période d'évaluation, vous DEVEZ m'envoyer une contribution (je suggère \$10 ou équivalent). Le source est disponible, ajoutez simplement \$10 de plus. Mon adresse se trouve à la fin de ce document.

Comme toujours, je dois mentionner que je ne fais AUCUNE garantie que le programme, bien que beaucoup testé, ne contient aucun bug, et donc, si un problème survient et cause des dommages directs ou indirects, JE PEUX ETRE TENU RESPONSABLE. VOUS UTILISEZ AREXXSUPERVISOR A VOS RIQUES ET PERILS.

1.3 Crédits et remerciements

ARexxSupervisor a été écrit sur un Amiga 500 (OS 2.04), avec le merveilleux Devpac 3 d'Hisoft.

ARexx est copyright © 1987 par William S. Hawes.

La géniale reqtools.library est Copyright © Nico François (elle n'est pas fournie dans cette archive mais vous pouvez la trouver partout dans les collections du domaine public).

L'interface graphique utilisateur a été créée avec GadToolsBox 2.0 (disponible sur la Fish 852), qui est ©1991-93 Jaba Development, optimisée et adaptée par moi-même.

A partir de la V1.10, ARexxSupervisor utilise l'errmsg.library, qui est copyrightée par moi-même. Vous pouvez la trouver dans les collections DP.

Remerciements et salutations à:

- Serge HAMMOUCHE de France-Festival-Distribution
- Commodore et Jay Miner pour avoir conçu l'Amiga ;-)
- Yves QUIQUEMPOIS
- François JACOBS
- Philippe DUSSART

1.4 Configuration nécessaire

Pour fonctionner, ARexxSupervisor a besoin de:

- Kickstart V37 ou plus récent (OS 2.04), il ne marchera pas sur le vieil OS 1.3. La seule chose à faire dans ce cas est... VOUS METTRE A JOUR!
- reqtools.library V38+ ©Nico François, qui peut être trouvé presque partout (pas Nico, la library :-)
- errormsg.library V1.0+, ©F.Delacroix, scrutez les bibliothèques DP.
- ARexx. Normalement fourni sur la disquette Workbench 2.0. Notez que vous devez lancer REXXMAST AVANT ARexxSupervisor (ce qui est normalement fait dans la startup-sequence 2.0)
- l'amigaguide.library (que vous possédez, puisque vous lisez ce fichier). On peut la trouver sur la Fish 870. Cependant ce n'est pas une erreur fatale de ne pas avoir installé cette bibliothèque, l'option d'aide en ligne sera simplement désactivée.
- Quelques connaissances sur la programmation et le debugage des interfaces ARexx !

1.5 Qu'est-ce qu'ARexxSupervisor

ARexxSupervisor est principalement un outil pour les développeurs d'entre vous qui voulez intégrer et tester/debuguer correctement des interfaces ARexx de leurs programmes.

ARexxSupervisor permet de contrôler à la souris les données qui sont envoyées à un port ARexx valide. Faites attention toutefois, c'est facile de faire des actions destructives, ou de provoquer des blocages ou gurus.

1.6 Lancer ARexxSupervisor

```
ARexxSupervisor peut être lancé à partir du
CLI
ou du
Workbench
.
```

1.7 Lancer ARexxSupervisor à partir du CLI

L'usage du CLI est: ARexxSupervisor [PUBSCREEN <name>]
PUBSCREEN/K

Ceci signifie que vous pouvez faire apparaître la fenêtre d'ARexxSupervisor sur n'importe quel écran public déjà ouvert. Exemple: ARexxSupervisor PUBSCREEN DEVPAC.1 ou ARexxSupervisor PUBSCREEN=DEVPAC.1 forcera

ARexxSupervisor à utiliser l'écran de Devpac 3. Par défaut, il utilise l'écran public par défaut (normalement l'écran du Workbench). Si l'écran public n'est pas ouvert ou est dans l'état privé, ARexxSupervisor "retombe" sur le défaut.

1.8 Lancer ARexxSupervisor à partir du Workbench

Vous lancez ARexxSupervisor à partir du Workbench en double ←
 -clicquant
 simplement son icône. Actuellement, un tootype est défini:
 PUBSCREEN=<name>. Voyez l'
 usage du CLI
 pour plus d'informations.

1.9 La fenêtre d'ARexxSupervisor

Quand ARexxSupervisor débute, il ouvre une grande fenêtre (←
 qui devrait
 quand même tenir sur un écran NTSC), avec une belle intergace GadTools
 faite d'un tas de gadgets. Ceux d'entre vous qui sont déjà familiers avec
 ARexx reconnaîtront tous les champs d'une structure REXXMSG, plus quelques
 gadgets auxiliaires. Voyons-les individuellement:

?
 Type
 Result
 Aide
 Extens.
 String
 I
 Entrée
 Token
 Q
 Sortie
 Envoie message
 Halte
 ARGx
 (x=0...F)

Result1

Trace globale

NoIO

Result2

Port

Remarquez que la plupart de ces gadgets ont un caractère ←
souligné,

indiquant que leur fonction est accessible par le clavier en pressant la touche correspondante. Des exceptions à cette règle sont le gadget d'icônification (qui n'a pas de raccourci clavier), et le gadget Help, qui est accessible par la touche Help. Bien sûr, pour que ces raccourcis marchent, il ne faut pas qu'un gadget de chaîne soit actif. Une autre option est la touche Help, qui, quand on la presse alors qu'un gadget de chaîne est actif (excepté les gadgets de chaîne Port et FileExt), amène un file requester. Le contenu de ce file requester est alors copié dans le gadget.

1.10 Gadget ? / Touche ? : Information

Rien de bien alarmant: cela amène le requester d'information, montrant quelques choses importantes comme la notice de Shareware et mon adresse.

1.11 Gadget Aide / touche Help: Affiche le fichier d'aide

Ce gadget affiche ce fichier (mais c'est peut-être grace à cela ←
que vous le

lisez maintenant!). Pour que cette option marche correctement, il vous faut avoir installé amigaguide.library dans le tiroir LIBS: (voyez

CONFIGURATION NECESSAIRE
) ,sinon il n'apparaîtra pas.

Le fichier que vous lisez maintenant doit s'appeler ARexxSupervisor.guide et se trouver dans le chemin d'accès d'AmigaGuide (qui peut être mis à jour grâce à la commande SetEnv, voyez votre documentation d'AmigaGuide). Je suggère que vous le copiez dans votre directory HELP:. Si le fichier est affiché avec succès par AmigaGuide, le programme est mis en sommeil jusqu'à ce que toutes les fenêtrre d'AmigaGuide soient fermées.

1.12 Gadget I : Iconification

Ce gadget sert à iconifier le programme quand vous trouvez qu'il prend trop de place sur votre écran. Vous avez bien sûr besoin de la workbench.library pour le faire. Le résultat sera la fermeture de la fenêtrre principale et l'apparition d'un nouvel AppIcon sur l'écran du workbench. Si une erreur survient lors de la création de l'AppIcon, la fenêtrre ne se fermera pas. Quand vous voulez que le programme revienne,

double-cliquez l'AppIcon et un requester apparaîtra vous demandant si vous voulez réveiller le programme, quitter ou annuler l'opération. Cliquez sur Wake up et la fenêtre réapparaîtra. Si elle ne peut pas (manque de mémoire...), vous devrez attendre encore un peu.

Si vous choisissez de quitter, le programme se terminera, à moins qu'il n'attende encore qu'ARexx réponde à un message précédemment envoyé (vous ne voulez pas réveiller le guru n'est-ce pas ? :-)

De plus, si ARexxSupervisor attend une réponse, il se réveillera automatiquement quand il la recevra. Voilà tout.

1.13 Gadget Q / touche Q / Gadget de fermeture: Quitter le programme

Dois-je vraiment vous expliquer cette fonction? ARexxSupervisor se terminera, à moins qu'il n'attende encore une réponse, auquel cas un requester apparaîtra, vous donnant l'occasion d'icôner plutôt.

1.14 Gadget Halte / touche H: Stoppe l'exécution de tous les programmes ARexx

Cette fonction est absolument identique à la commande HI que vous taperiez dans un Shell: tous les scripts ARexx s'arrêteront (à moins que cette exception ne soit détournée) et enverra un message "execution halted".

Notez que ceci doit constituer votre DERNIERE chance d'arrêter un programme devenu incontrôlable. En effet, arrêter tous les programmes n'est pas très poli; vous pourriez détruire des données importantes pour d'autres processus.

1.15 Gadget Trace globale / touche G: force le mode trace pour tous

C'est une CheckBox, donc elle a deux états: activée ou non. Quand elle n'est pas sélectionnée, tous les programmes ARexx tournent normalement. Quand elle l'est, tous les programmes ARexx entrent en mode de traçage interactif. Pour que l'exécution d'un programme continue, vous devez presser RETURN dans sa console de sortie (ce DOIT être une console interactive) ou dans la console de traçage globale si elle est ouverte. Vous pouvez alors examiner les variables... pour voir ce qui ne va pas dans votre programme.

Mais encore une fois, n'oubliez pas que cette fonction affecte l'environnement tout entier, et doit donc être utilisée avec prudence.

Note: vous obtiendriez le même résultat en lançant le programme TS ou TE pour l'arrêter.

1.16 Gadget Port / touche P: nom du port de destination

Entrez ici le nom du port à adresser pour l'envoi du message. Ce nom EST sensible à la "majusculté" (!). Le port n'a pas besoin d'être ouvert à ce moment, mais il devra l'être quand vous enverrez le message. Ce sera par exemple AMIGAGUIDE.1 pour la première occurrence d'amigaguide. Aussi, faites attention à ce que vous envoyez à qui. Normalement, par exemple, seul ARexx est capable de comprendre des messages comme l'ouverture de la console de traçage. C'est très facile d'envoyer des messages incompréhensibles! Le port où vous pouvez dialoguer avec le processus ARexx principal se nomme REXX.

1.17 Gadget Type / touche Y: type du RexxMsg

Ce gadget cyclique contrôle le type du Message à envoyer. Ces types sont: ↔

Envoie une commande

Ajouter un clip

Appeler une fonction

Retirer un clip

Fermer console trace

Ajouter une biblio.

Ouvrir console trace

Retirer une biblio.

Au moment de l'envoi, le champ Type et les modifier flags seront ↔ couplés par

un OU. Normalement, seuls Send command et Invoke function peuvent être envoyés à d'autres hôtes que REXX.

1.18 Envoie une commande

Ceci signifie que le message est une commande à exécuter par l'hôte. ARG0 contient normalement le nom de la commande (qui est normalement le nom d'un fichier pour REXX) suivi d'une chaîne d'arguments. ARG1 à ARGF sont inutilisés.

Le modifier "Token" force cependant ARexx à "tokeniser" la chaîne entière, qui devient alors autant de chaînes d'arguments.

"NoIO" empêchera le nouveau processus d'hériter des canaux IO standards.

Le modifier "String" dira à REXX que c'est un "fichier chaîne" plutôt qu'un nom de fichier.

Le modifier "Result" demandera à l'hôte de renvoyer un résultat pour cette commande (les commandes ne renvoient généralement pas de résultat).

1.19 Appelle une fonction

Le message demandera à l'hôte de faire une action, de laquelle il devrait renvoyer un résultat. Le nom de la fonction est en ARG0 et les arguments de ARG1 à ARGF.

On réclame un résultat en sélectionnant le modifier flag correspondant.

Les modificateurs "String" et "NoIO" ont le même effet que pour l'envoi de commandes.

Le résultat sera placé dans Result2 si il n'y a pas d'erreur (si Error1 est 0).

1.20 Ferme la console de traçage

Ce message fermera la console de traçage globale si elle était ouverte.

Il n'a de signification que pour le port REXX.

1.21 Ouvre la console de traçage

Ceci ouvrira la console de traçage, et tous les programmes qui sont en mode trace enverront leur informations de traçage vers cette fenêtre au lieu de leurs canaux StdErr ou StdOut (qui peuvent ne pas être interactifs!). Cependant, ça peut être un peu la pagaille si plusieurs programmes sont en mode trace à la fois.

A envoyer uniquement à REXX.

1.22 Ajoute une entrée au clipboard

ARexx entretient un "carnet de notes" qui est partagé par tous les processus. Ceci ajoute une entrée à cette liste. ARG0 est le nom de l'entrée, ARG1 est sa valeur. ARG2 est automatiquement rempli avec la longueur de ARG1. C'est une restriction de l'utilisation d'ARexxSupervisor: vous ne pouvez pas mettre n'importe quel type de données dans le carnet de notes, vous devez utiliser une chaîne. J'espère que ça ne posera pas de problème.

A envoyer uniquement à REXX.

1.23 Enlever une entrée du clipboard

Ceci retire simplement une entrée du carnet de notes. ARG0 est le nom de l'entrée à retirer.

Pour REXX seulement.

1.24 Ajoute une bibliothèque

Ce type ajoute une bibliothèque de fonctions à la liste interne d'ARexx, pour que les programmes puissent accéder à ses fonctions.

ARG0 est le nom de la bibliothèque (ex:amigaguide.library),

ARG1 est la priorité dans cette liste. Ce doit être un nombre entre -100 et 100. (ex: 0)

ARG2 est l'offset d'interrogation pour la bibliothèque. Pour amigaguide, c'est -30.

ARG3 est la version réclamée de la bibliothèque. 0 acceptera n'importe quelle version.

Ce message est pour REXX seulement.

1.25 Retire une bibliothèque

Ce message retirera bien entendu la bibliothèque spécifiée de la liste des bibliothèques interne à ARexx.

ARG0 est le nom de la bibliothèque à retirer. ex:amigaguide.library.

1.26 Gadget FileExt / touche X: Extension de fichier

C'est l'extension de fichier qui sera ajoutée à un nom de fichier lors de la recherche d'un programme ARexx. Cette méthode sert à différencier les programmes écrits pour des hôtes différents (.rexx,.editor, etc...)

1.27 Gadget StdIn / touche é: Canal d'entrée standard

C'est le canal qui sera utilisé pour l'entrée. Si ce n'est pas un nom de fichier valide (comme CON:///Hello), alors StdIn sera le même canal que StdOut. Et si StdOut n'est pas valable non plus, NIL: sera utilisé. N'oubliez pas que ce canal est TOUJOURS ouvert par ARexxSupervisor en MODE_OLDFILE, alors que StdOut est TOUJOURS ouvert en MODE_NEWFILE. Alors faites attention.

Si vous appuyez sur Help alors que ce gadget est actif, un filerequester apparaîtra.

1.28 Gadget StdOut / touche O: Canal de sortie standard

Voilà un autre canal: celui qui sera utilisé pour les ↔ sorties.

Attention, si vous voulez utiliser les options de traçage d'ARexx, vous devez fournir un canal interactif, sinon vous risquez le blocage. Pour plus de sécurité, vous pouvez ouvrir la console globale de traçage. Voyez la description du gadget

StdIn
pour plus d'informations.

Si vous pressez Help alors que l'un de ces gadgets est actif, un file requester apparaît.

1.29 Gadgets ARGx / touches 0..F: Chaînes arguments

Ces 16 gadgets de chaîne sont les arguments qui seront utilisés pour passer le message à l'hôte. Leur utilisation varie beaucoup avec le type de message, mais les cas où les 16 slots sont utilisés sont très rares. En fait, cela ne peut arriver que pour les fonctions.

Si vous pressez Help alors que l'un de ces gadgets est actif, un file requester apparaît.

1.30 Gadget NoIO / touche N: modifier flag NoIO

Quand il est mis, ce modifier flag supprime l'héritage des canaux IO pour le processus fils (StdIn et StdOut sont ignorés).

1.31 Gadget Result / touche R: modifier flag Result

Ce flag doit être mis lorsque vous voulez que la commande ou la fonction que vous invoquez vous retourne un résultat.

1.32 Gadget String / touche S: modifier flag String

Ce modifier dit à ARexx que la chaîne dans ARG0 n'est pas le nom d'un programme mais le programme lui-même (c'est un "fichier-chaîne").

1.33 Gadget Token / touche T: modifier flag Token

Ce flag dit à ARexx d'analyser ARG0 pour que la commande ne reçoive pas une mais plusieurs chaînes d'arguments.

1.34 Gadget Send message / touche M: Envoie le message à l'hôte

```

Ce bouton créera réellement la structure REXXMessage et l'enverra ←
à l'hôte
dont le nom est donné dans le gadget
Port
.

```

Tous les champs sont remplis et le message est envoyé, ARexxSupervisor interdit tous ses gadgets à l'exception de l'iconification, l'aide, l'information, Halte, le traçage et Quit. Il est cependant impossible de quitter le programme tant qu'on n'a pas répondu au message. En général, ARexx répond quand le programme est terminé. ARexxSupervisor actualise alors les gadgets de résultats, autorise à nouveau ses gadgets et reprend les opérations normales.

1.35 Gadget Result1: Code de retour principal

C'est le code d'erreur primaire renvoyé par la fonction précédemment exécutée. Généralement, une valeur non nulle indique un code d'erreur ou un niveau de sévérité. Si Result2 est vide, c'est le code de retour de la fonction, affiché par: Return Code: %ld. Si Result2 n'est pas nul, alors c'est un niveau de sévérité d'erreur, affiché par: Severity level: %ld, et Result2 contient le code d'erreur: Error code: %ld.

Si Result1 est nul, Result2 peut contenir le résultat de la fonction ou de la commande que vous aviez demandé en positionnant le modifier flag Result (suivez toujours?:-)

La valeur de ce gadget peut être trouvée dans la variable d'environnement nommée AResult1.

1.36 Gadget Result2: Code de retour secondaire.

```

C'est le résultat secondaire de l'appel précédent. ←
Pour plus
d'informations voyez le gadget
Result1
.

```

La valeur de ce gadget est copiée dans la variable d'environnement nommée AResult2.

1.37 Traduction

A partir de la V1.10, ARexxSupervisor est complètement localisé, ce qui signifie que vous pouvez le faire fonctionner dans votre langue pourvu que le bon fichier catalogue soit disponible dans votre tiroir LOCAL:Catalogs/<lang>. Pour l'instant, il n'y a qu'un catalogue français, alors si vous voulez un catalogue dans votre langue, vous devrez le faire vous-même. La seule chose que vous avez à faire est d'éditer le fichier nommé ARexxSupervisor.ct et de remplir les blancs avec vos traductions des chaînes placées en commentaire. Envoyez-moi alors le fichier obtenu pour que je puisse générer un fichier catalogue que je vous renverrai. Le fichier catalogue sera inclus dans la distribution suivante (avec les bons crédits bien sûr). Pensez aussi à traduire le fichier guide !

1.38 Améliorations futures

ARexxSupervisor est bien entendu sujet à des améliorations. Si vous pensez à quelque chose que vous voudriez voir dans la prochaine version, faites-le moi savoir. En voici qui me viennent à l'esprit:

- une option auto-RexxMast
- un port ARexx !
- tout ce que vous pourrez suggérer !

1.39 Contacter l'auteur

SVP n'oubliez pas qu' ARexxSupervisor est SHAREWARE. Si vous le trouvez utile, envoyez-moi une petite contribution (je suggère \$10 ou équivalent) à l'adresse ci-dessous.

Frédéric DELACROIX
5 rue d'Artres
59269 QUERENAING
FRANCE, TERRE

Aussi, toutes les suggestions/rapport de bug/cartes postales sont les bienvenues.
