**Wikiprint Book**

**Title: fr:Documentation/ARexxAPI**

**Subject: YAM - Yet Another Mailer - fr:Documentation/ARexxAPI**

**Version: 13**

**Date: 12/02/2015 06:22:42 AM**

## Table of Contents

Autres langues: 3
Référence - Interface ARexx 3

**Autres langues:**

- English
- Español
- **Français**
- @10 - @27

## Référence - Interface ARexx

À l'aide de scripts ARexx vous pouvez ajouter de nouvelles fonctions à YAM ou bien lui laissé effectuer des fonctions automatiquement. YAM offre un ensemble de commandes ?, qui peuvent être appelées à travers le port ARexx « YAM ». Il y aura des explications tout au long des sous-chapitres en utilisant le format suivant :

**NOM**

Le nom de la commande, une courte description de ce qu'elle fait et la version de YAM depuis laquelle elle est implémentée.

**SYNTAXE**

Les arguments et options acceptés par la commande. La syntaxe utilise des caractères spéciaux pour indiquer le type particulier de l'argument attendu, en suivant le style de l'AmigaOS :

/A Le paramètre est facultatif
/K Le paramètre doit être précédé du mot clé
/N Argument ou résultat numérique
/M L'argument ou le résultat est une liste composée (de zéro à plus) éléments
/S Le paramètre fonctionne comme un interrupteur ; il est activé quand il est mentionné.

**FONCTION**

Donne une description de la commande.

**ENTREE**

Décrit en détail les paramètres acceptés par la commande. Prenez garde lorsque vous passez des arguments contenant des espaces. Par exemple :

```
sub = 'Hello World'
'WRITESUBJECT' sub
```

ne fonctionnera pas. Il aurait fallu écrire :

```
'WRITESUBJECT '"sub""
```

ou

```
'WRITESUBJECT "Hello World"'
```

Veuillez noter qu'à cause de l'utilisation en interne de la fonction ReadArgs(), l'hôte ARexx nécessite l'échappement de certains caractères spéciaux comme un retour à la ligne (0x0a) ou le caractère d'échappement (0x1b) si vous voulez qu'il soit inclus dans votre chaine finale, dans le cas contraire, il sera éliminé par la fonction ReadArgs().

Cela signifie que vous devez utiliser les séquences d'échappement suivantes dans vos chaines de caractères :

*N remplace 0x0a
*E remplace 0x1b
** remplace *
*" remplace "

Par exemple la commande suivante écrit un chaine de caractère contenant un retour à la ligne dans l'éditeur :

'WRITEEDITOR "TEXT Bonjour Joe,*N je voudrais vous rencontrer."'

Si un paramètre représente un nom de fichier, vous devez garder à l'esprit que les noms de fichier sous AmigaOS peuvent contenir des espacements ou autres caractères spéciaux qui ferant qu'ARexx traitera le nom comme plusieurs mots au lieu d'un seul mot. De ce fait, un nom de fichier devrait **toujours** être entouré de guillemets ("), qu'importe qu'il contient des espaces ou pas. Les scripts peuvent facilement être testés pour ce cas en procurant un fichier du disque RAM; par exemple "RAM Disk:/foo/bar".

**RESULTAT**

Retourne un résultat attendu. Les commandes peuvent retourner un résultat de trois manières différentes. Jetez un oeil aux *exemples* qui utilisent la commande FOLDERINFO :

```
FOLDERINFO

    -> RESULT = "0 Reçus incoming 10 2 4 23030 1"

FOLDERINFO VAR fi

    -> fi = "0 Reçus incoming 10 2 4 23030 1"

FOLDERINFO STEM fi.

    -> fi.number = 0

        fi.name = "Reçus"
        fi.path = "incoming"
        fi.total = 10
        fi.new = 2
        fi.unread = 4
        fi.size = 23030
        fi.type = 1
```

Un autre exemple qui retourne un résultat de type /M :

```
ADDRFIND STEM found. "Marcel Beck" NAMEONLY

    -> found.alias.count = 2

        found.alias.0 = "Mars"
        found.alias.1 = "mbe"
```

**ATTENTION**

Toutes sortes d'informations auxquelles il faut prêter attention en utilisant cette commande.

**NOTES**

Remarques variées sur la commande.

**EXEMPLE**

Un bout de code ARexx pour illustrer l'utilisation de la commande.

**BUGS**

Problèmes connus ou corrigés avec cette commande.

**VOIR AUSSI**

Liens relatifs à d'autres commandes.

---


- Liste des commandes ARexx par ordre alphabétique ?
- Liste de toutes les commandes ARexx triées par thème ?