ClassAction

Gasmi Salim

COLLABORATORS			
	TITLE : ClassAction		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Gasmi Salim	March 26, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

ClassAction

1.1 Guide de ClassAction 3.1

```
                    ClassAction
                    Version 3.1


Qu'est-ce que ClassAction ?     Lisez tout d'abord ceci
Exigences système               Ce dont vous avez besoin

Installation                    Comment installer ClassAction
Après l'installation            Installation Partie II

Utiliser ClassAction            Comment utiliser cet outil

Configuration                   Changer les ToolTypes
ClassAction Prefs               Configuration des classes et des actions

Les commandes ARexx             Le port ARexx et les commandes

Améliorations futures           Ce qui devra être rajouté
Historique & Fonctionnalités    Depuis le début...
Enregistrement                  Pourquoi et comment s'enregistrer

    Licence     L'auteur     Remerciements      Bugs connus
```

1.2 Qu'est-ce que ClassAction ?

```
ClassAction est un petit utilitaire qui va simplifier la vie
de tous les utilisateurs de disque dur.

Lorsque vous possédez un disque dur, vous avez forcément une tonne de
fichiers :  exécutables, modules, images, sources, animations, sons...

ClassAction détermine pour vous le type de fichier sélectionné
et propose une liste d'actions a effectuer sur ce fichier.

Par exemple, si vous sélectionnez une image GIF, elle sera reconnue par
```

ClassAction comme étant de la classe GIF, et une liste d'action sera affichée avec des actions telles que 'Afficher' ou 'Editer'.

ClassAction est fortement configurable, vous pouvez ajouter vos propres classes et actions.
Les actions utilisent des programmes externes, vous pouvez donc utiliser votre afficheur de Gif préféré (ou n'importe quoi d'autre).

ClassAction a une AppIcone, un port ARexx, est localisé et est une commodité.

ClassAction utilise la xfdmaster.library pour décompresser automatiquement les fichiers compressés.
Avec cette possibilité, même les classes compressées peuvent être reconnues.

ClassAction utilise très peu de mémoire et de temps CPU.

Si vous prenez un peu de temps à configurer ClassAction, vous pourrez faire n'importe quoi avec !!! C'est une façon simple de gérer ses fichiers.

Avant de décider de le virer, essayez le !!!!!!

Si vous voulez en savoir plus sur ClassAction, cliquez içi

1.3 Un peu plus sur ClassAction

ClassAction et ClassActionPrefs sont (C) 1994-96 par Gasmi Salim

Ce logiciel est distribué en tant que ShareWare. Vous êtes libre de l'utiliser !!! et de le diffuser tant que vous ne modifiez aucun des fichiers de l'archive.
MAIS si vous l'utilisez régulièrement, vous devez vous enregistrer !!!

Les distributeurs de DP sont autorisés à inclure le package de ClassAction dans leur collection tant qu'ils me préviendront qu'ils l'ont inclu.

Ceci est la version 3.1 de ClassAction.
Si vous la trouvez sympa, vous DEVEZ vous Enregistrer

J'espère que vous trouverez cet outil utile.
(au moins, je le trouve utile :])

Pour des informations techniques sur ClassAction :
Informations Techniques

1.4 Exigences système

Pour utiliser ClassAction, vous avez besoin des trucs suivants :

- Amiga OS 2.0 ou supérieur
- MUI 3.0 ou superieur
- un disque dur
  (ClassAction sans intérêt sans disque dur)

That's all folks !!!

1.5 Configuration de ClassAction

L'interface et les fonctionnalités de ClassAction sont configurables via les ToolTypes.

Pour changer une valeur d'un ToolType, sélectionner simplement l'icone de ClassAction et utiliser l'option 'Information' dans le menu 'Icones' du Workbench.

Voici une liste de tous les ToolTypes :

Note : La valeur par défaut d'un ToolType est utilisée lorsque le ToolType n'est pas trouvé.

==========

DONOTWAIT

N'enlevez pas ce ToolType, il est nécessaire si ClassAction est lancé depuis la WBStartup.

==========

DECRUNCH

Si vous mettez ce ToolType à YES (i.e. DECRUNCH=YES), ClassAction essayera de décompresser les fichiers compressés, grâce à la xfdmaster library.

Cela est nécessaire si vous souhaitez reconnaitre les fichiers compressés.

Bien sûr, dans ce cas, le fichier sera pré-chargé en mémoire et un buffer sera alloué pour les données décompressées.
Ceci consomme donc de la mémoire (environ 2.5 x la taille du fichier).
Donc si vous n'avez pas suffisemment de mémoire, vous ne devriez pas utiliser cette possibilité.

Valeur par défaut : NO

==========

ICONFILE

Ce ToolType est le fichier utilisé pour l'AppIcone.
De cette façon, vous pouvez utiliser n'importe quel icone (image)
pour ClassAction.

Note : Ne mettez pas d'extension .info au nom du fichier icone.

Exemple : ICONFILE=Sys:icons/head
utilisera l'icone head.info du répertoire Sys:Icons/
comme AppIcon.

Si ClassAction ne parvient pas à charger votre icone, il
utilisera celui par défaut (ClassAction.info).

Valeur par défaut : "" (Utilise l'icone par défaut)

==========

CLISIZE

C'est la taille de la fenêtre de sortie CLI, lorsque SystemTags()
est utilisé.

La syntaxe est : DEVICE:TopX/TopY/Width/Height/Title

De cette façon, vous pouvez utiliser une autre unité à la place
de CON: , ou définir de nouvelles position et
dimensions pour la fenêtre de sortie CLI.

ATTENTION :

Si ce ToolType comporte une valeur erronnée, ClassAction ne
n'ouvrira pas de CLI.
NE modifiez pas cette valeur à moins que vous ne sachiez ce que
vous faites.

** NE JAMAIS ** ajouter de commandes telle que AUTO, CLOSE, WAIT
à cet argument car ClassAction le fait automatiquement.

** NE JAMAIS ** mettre d'espaces dans le titre.

Dans la version 2.0, un ToolType similaire : OUTPUT était utilisé.
Il n'est plus nécessaire actuellement.

Valeur par défaut :
CON:0/0/640/100/ClassAction_Output_Window

==========

GTLIKE

Si vous declarez simplement ce ToolType , alors le selecteur
de fichier MUI ressemblera aux anciens de Gadtools , c'est a dire
que :

Les repertoires seront en blanc.
Les fichiers en noir.
Et que les .info seront affiches.

==========

CAPREFS

Ce ToolType contient la chemin de ClassActionPrefs et est
necessaire au lancement de ClassActionPrefs a partir du boutton Prefs.

Valeur par défaut : SYS:prefs/ClassActionPrefs

==========

DRIVE1 à DRIVE50

Mettez ces ToolTypes à un chemin valide et ils seront affichés
dans les boutons du Requester de fichier.
C'est utile pour aller rapidement vers un répertoire.

La syntaxe est :

DRIVEx=<Texte du Bouton>,<Chemin>

i.e. DRIVE9=Jpeg,dh0:gfx/pictures/jpeg

De cette façon, le texte du bouton 9 sera 'Jpeg', mais le chemin
sera dh0:gfx/pictures/jpeg.

Vous pouvez également fournir uniquement le <Texte du Bouton>
sans le <Chemin>.
i.e. DRIVE3=dh0:libs

Dans ce cas, le texte du bouton sera le même que le chemin.
Mais dans ce cas, n'utilisez pas de chemin trop long car ils ne
seront pas affichés dans les petits boutons.

Valeur par défaut : "" (rien)

1.6 ClassAction Prefs

La définition de nouvelles classes et actions est le coeur du programme.
Pour faire ceci, il faut utiliser ClassActionPrefs.

L'utilisation de ClassActionPrefs devrait être simple...

Alors allons-y !!

La fenêtre est divisée en deux parties : les Classes et les Actions.

D'abord, choisissez une classe, les actions associées seront affichées dans la partie des actions.

Pour ajouter ou supprimer une classe, cliquez simplement sur le bouton correspondant.

C'est la même chose pour les actions...

1. Qu'est-ce qu'une classe
2. Créer une nouvelle classe
3. Définir une nouvelle action
4. La fonction Apprendre

1.7 Utiliser la fonction apprendre

La fonction Apprendre est destinée a vous aider a definir de nouvelles classes.

Quand vous definissez une nouvelle classe, il vous faut parfois aussi definir des offsets pour la classe, et cela peut étre long et ennuyeux d'editer les fichiers pour essayer de deviner quels pourraient bien étre les offsets .

C'est pour cette raison que le merveilleuse founction Apprendre a eté écrite.

Pour l'utiliser, suivez cette procedure :

1 : definissez votre classe, remplissez le nom.

2 : clicker sur le bouton 'Apprendre'.

    Choisissez l'efficacite de l'apprentissage (de 75 a 100%)
    Plus ce nombre sera elevé meilleurs seront les resultats.
    Je vous conseille de toujours utiliser 100%

3 : Selectionner autant de fichiers que vous voulez avec le
    requester qui apparait Tant qu'ils appartiennent tous a
    la classe que vous voulez definir.
    Plus vous selectionnerez de fichiers, meilleur sera le
    resultat.

    ClassAction va essayer de trouver les offsets !
    (Pour selectionner plusieurs fichiers, utilisez la touche shift)

4: Apres analyse, vous avez les offsets trouvés.

5: Modifiez les a la main si neccessaire.

6: clickez sur 'Accepter' pour les utiliser pour votre classe
ou clickez sur 'Annuler' pour annuler.

ATTENTION :

Pour que ceci fonctionne , il faut imperativement étre sur que :

* TOUS LES FICHIERS CHOISIS SOIENT DE LA MEME CLASSE.

* TOUS LES FICHIERS CHOISIS SOIENT NON CRUNCHES.

C'est tout !! facile non ?

1.8 Qu'est-ce qu'une classe

Une classe est une famille de fichiers. Par exemple, l'ensemble des sources C peut être considéré comme une classe, appelons la Classe C.

Avec ClassActionPrefs, vous pouvez définir autant de classes que vous le souhaitez, tant que vous décrivez la façon de la reconnaitre.

Pour décrire comment reconnaitre une classe, il y a deux méthodes : le nom du fichier (motif ou match name) et l'analyse du contenu du fichier.

L'attribut MatchName est utilisé pour reconnaitre un fichier en fonction de son nom.
Les attributs Offsets sont utilisés pour reconnaitre un fichier en fonction de son contenu.

Il y a trois classes internes que vous ne pouvez pas supprimer, elles sont affichées en blanc dans la liste des classes.

La première est appelée "Classe inconnue".
Cette classe contient tous les fichiers que ClassAction ne peut pas reconnaitre.

La seconde est appelée "Action génériques"
Cette classe contient des actions qui seront affichées pour TOUTES les autres classes...

Intérêt : si vous voulez avoir une action 'Copier' pour toutes les classes, vous pouvez la définir pour toutes les classes existantes, mais looooong et ennuyeux.
Un façon plus appropriée de faire ceci est de rajouter cette action dans la liste des "Action génériques". 'Copier' sera alors affiché pour

toutes les classes.

Les actions génériques sont affichées en blanc dans la liste des actions de ClassAction et ne sont visibles que lorsque la fenêtre de ClassAction est ouverte.
Elles ne sont pas accessibles lorque ClassAction est AppIconifié, ou via la commande ARexx : Load.

La troisieme classe est appelée "Tiroir"
Cette classe contient les actions disponibles quand vous selectionnez un tiroir .

1.9 Créer une nouvelle classe

Cliquez simplement sur le bouton 'Ajout' dans la partie Classes pour ajouter une nouvelle classe.

Une classe a trois propriétés :

- un nom
- un motif
- des offsets

o Le nom est simplement le nom de la classe, c'est à vous de le choisir.
ATTENTION : un nom de classe doit être unique.

o Le motif est une expression régulière AmigaDos, telle que :

#?.c , mod.#? , #?.c|#?.h , #?b[a|c] , #?toto? ....

(lisez le manuel AmigaDOS pour tous les caractères génériques)
ATTENTION : N'utilisez pas le caractère * mais plutôt #? .

Le motif ne différencie pas minuscules et majuscules, donc toto.C peut correspondre à #?.c

Si vous vous définissez une classe grâce aux motifs, il faut être sûr que cette définition est toujours correcte.

Exemple : si vous définissez la classe GIF avec le motif #?.gif tous les fichiers terminant par .gif seront reconnus comme des images GIF.

Mais êtes vous certain que tous les fichiers .gif sont des images GIFs, ou que toutes vos images GIF dispose de l'extension .gif ?

En fait, il ne faut utiliser les motifs que pour ces deux situations :

- il y a une bijection entre le motif et les noms des fichiers de

la classe
(ex: #?.info est un bon motif pour la classe des Icones)

- vous n'avez pas le choix
(ex: comment reconnaitre un source C, à part avec #?.c)

o Dans les autres cas, il vaut mieux utiliser les Offsets.

Un Offset est un endroit dans le fichier où l'on doit trouver quelque chose de particulier pour l'identifier.

Par exemple, les images GIF commencent toujours par la chaine 'GIF' à l'Offset 0.

Il y a trois syntaxes pour définir les Offsets :

==========

Syntax n\textdegree{}1 : Offset,ChaineHexa

Offset est un nombre DECIMAL contenant l'Offset.
ChaineHexa est une chaine HEXADECIMALE qui doit se trouver à cet Offset.

Exemple 1 : 0,4f4a signifie que le fichier doit commencer avec les octets $4f et $4a

Exemple 2 : 9,448b3c signifie que, à l'octet n\textdegree{}9 doit se trouver : $44 $8b $3c

==========

Syntax n\textdegree{}2 : Offset,'Chaine'

Offset est un nombre DECIMAL contenant l'Offset.
Chaine est une chaine ASCII qui doit se trouver à cet offset.

Exemple 1 : 0,'GIF' signifie que le fichier doit commencer par la chaine 'GIF'
Exemple 2 : 9,'FuBar' signifie que, à l'octet n\textdegree{}9 doit se trouver la chaine 'FuBar'

==========

Syntax n\textdegree{}3 : Offset,"Chaine"

Offset est un nombre DECIMAL contenant l'Offset.
Chaine est une chaine ASCII qui doit se trouver à cet offset.

Notez la différence avec la syntaxe précédente, içi nous

```
    utilisons " pour définir la chaine, et dans la syntaxe n\textdegree ↩
       {}2,
    nous utilisions '.

    C'est le même concept que la syntaxe n\textdegree{}2, mais dans ce cas ↩
       ,
    il n'y a pas de différentiation entre minuscules et majuscules
    lors de la comparaison.

    Par exemple : un fichier AmigaGuide commence toujours par la
                  chaine @database en majuscules ou minuscules.

    Si vous utilisez la deuxième méthode pour reconnaitre les
    fichiers AmigaGuide, avec 0,'@database', ClassAction ne
    reconnaitra pas un fichier commencant par @DATABASE comme étant
    un AmigaGuide.

    Par contre, cela fonctionne si l'Offset est défini avec la
    syntaxe n\textdegree{}3 : 0,"@database"


==========================================================================


Vous pouvez préciser jusqu'à 5 Offsets pour définir une classe.
Un fichier est reconnu pour la classe si tous les Offsets
correspondent.

Ex : si la classe X est définie par :

Offset n\textdegree{}1 :  0,4a8b6c
Offset n\textdegree{}2 : 58,14

Tous les fichiers commencant par $4a8b6c ET ayant $14 à l'octet 58
appartiendront à X.

Pour définir plusieurs Offets, cliquez simplement sur le bouton
cyclique 'Offset #' pour activer le prochain Offset.


Remarque : La classe ASCII

Il existe une command interne pour les Offsets, nommée : ASCII[]

Si vous insérez cette commande dans l'Offset n\textdegree{}1
(i.e Offset#1=ASCII[]), elle va reconnaitre les fichiers ASCII.
ClassAction n'essayera cette commande que lorsque tout le reste
aura échoué.
Grâce à ceci, les documents Amigaguide (qui sont des fichiers ASCII)
ne seront pas interprétés comme ASCII si vous avez défini une classe
Amigaguide.

Normalement, vous ne devriez pas utiliser cette commande car j'ai
fourni un fichier de préférences standard qui contient déjà la
classe ASCII, définie grâce à cette commande.
```

1.10 Définir une nouvelle action

Une fois la classe définie, il faut y rajouter des actions.
Chaque classe peut contenir autant d'actions que vous le souhaitez.

Cliquez sur le bouton 'Ajout' dans la partie des actions pour ajouter
une action.

Une action a 6 propriétés :

- un nom
- un mode d'exécution
- une taille de pile (seulement si le mode d'exécution est CLI)
- un délai (seulement si le mode d'exécution est CLI)
- une commande à exécuter
- un Etat RescanDir vrai ou faux.

o Nom est le nom de l'action.

o Mode d'exécution peut être : Cli , WB , No Cli or ARexx .

o Commande est une ligne de commande AmigaDOS valide, qui peut
contenir des paramètres.
VOUS DEVRIEZ toujours utiliser un chemin absolu pour les
exécutables.

Exemple : utilisez C:Copy au lieu de Copy dans la ligne de commande.

Vous pouvez insérer dans la ligne de commandes, des Arguments
et des Commandes internes à ClassAction.

o Mettez le CheckMark RescanDir actif si vous voulez que cette action
rescanne le repertoire courant , c'est utile pour les actions qui
touchent au contenu du repertoire comme par exemple renommer
ou effacer un fichier.

Boutons

o Les boutons fleches Haut et Bas permettent de trier les actions.

o Le bouton 'Charg' vous demande de choisir un exécutable pour la
ligne de commandes.

o Le bouton 'Comm' affiche une fenêtre qui liste tous les arguments
ou commandes internes possibles pour la ligne de commandes.

o Vous pouvez copier une action deja existante sur celle selectionée
en cliquant sur le gadget a droite du nom de l'action.
Il apparaitra une liste arborescente , selectionnez la classe source
double cliquez dessus , et double cliquez sur l'action voulue;
Celle ci sera copiee dans l'action en cours.

1.11 Mode CLI

```
**** Mode CLI ****

Si 'Cli' est choisi, l'action sera lancée à partir d'un CLI et la taille
de la pile sera déterminée par la valeur donnée dans : Pile
(par défaut : 4096)

Ce mode d'exécution n'ouvrira de CLI que si cela est nécessaire (lorsque
l'exécutable affiche quelque chose).

Vous pouvez définir la propriété Délai pour ce mode :

Si Délai est négatif (i.e. Delai = -1), le CLI restera ouvert jusqu'à ce
que vous le fermiez grâce au gadget de fermeture dans le coin supérieur
gauche de la fenêtre.

Si Délai est nul (i.e Delai = 0), le CLI se fermera de lui-même dès que
la tâche se terminera.

Si Délai est positif (i.e. Delai = n avec n>0), le CLI attendra n
secondes avant de se refermer, mais vous pouvez forcer la fermeture
grâce au gadget correspondant.

La dimension du CLI utilisé est définié par le ToolType CLISIZE.
(l'ancien ToolType OUTPUT est maintenant obsolète).
```

1.12 Mode WB

```
*** Mode WB ***

Si 'WB' est choisi, aucun CLI se sera ouvert, et ClassAction simulera
une exécution de l'action depuis le Workbench.
L'outil sera exécuté avec comme arguments, ceux contenus dans son icône.

ATTENTION : ce mode est uniquement valable pour les exécutables qui
            disposent d'une icône.
```

1.13 Mode NO CLI

```
*** Mode NO CLI ***

Si 'No CLI' est choisi, aucun CLI ne sera ouvert, même si le programme
affiche quelque chose, mais la tâche est tout de même exécutée depuis
un CLI.
```

1.14 Mode ARexx

*** Mode ARexx ***

Si 'ARexx' est choisi, la commande rx est exécutée avec la ligne
d'exécution spécifiée. Cette ligne DOIT être un script ARexx.
Bien sûr, RexxMaster doit être actif et and Rx doit se trouver dans le
répertoire Sys:rexxc/ pour fonctionner.

1.15 Arguments

Actuellement 8 commandes Arguments sont disponibles :

les 4 premieres commandes sont en minuscules : [f] [s] [b] [x]
et inclusent le resultat entre guillemets .

[f] : Chemin complet du fichier selectionne avec guillemets
[s] : Chemin complet du fichier selectionne sans suffixe avec guillemets
[b] : Nom du fichier selectionne avec guillemets
[x] : Nom du fichier selectionne sans suffixe avec guillemets

Les 4 commandes suivantes font la meme chose mais sans les guillemets

[F] : Chemin complet du fichier selectionne
[S] : Chemin complet du fichier selectionne sans suffixe
[B] : Nom du fichier selectionne
[X] : Nom du fichier selectionne sans suffixe

Example : supposons que vous ayez choisi le fichier ram:env/sys.prefs

[f] = "ram:env/sys.prefs"
[s] = "ram:env/sys"
[b] = "sys.prefs"
[x] = "sys"

[F] = ram:env/sys.prefs
[S] = ram:env/sys
[B] = sys.prefs
[X] = sys

Example : supposons que vous avez choisi le fichier ram:main.c

* La ligne Exec
c:copy [f] [F].bak
sera remplacee par :
c:copy "ram:main.c" ram:main.c.bak

* La ligne Exec

```
            c:copy [f] [S].bak
  sera remplacee par :
            c:copy "ram:main.c" ram:main.bak
```

1.16 Commandes

```
Actuellement, cinq commandes sont possibles :


REQD[texte] : Demande un répertoire.
REQF[texte] : Demande un fichier.
REQV[texte] : Demande un volume.
REQT[texte] : Demande un texte.
SURE[texte] : Demande à l'utilisateur de confirmer.


=========================================================================

Commandes REQ


    Les commandes REQ affichent une requête Reqtools ayant pour titre :
    [texte]. Elle sont utiles lorsque vous avez besoin de comandes
    interactives.

    Exemple :
        bin:lha x [f] to REQD[Choisissez un répertoire où extraire]

    Ceci affichera une requête de répertoire vous permettant de choisir
    le répertoire cible. Le fichier selectionné [f] sera alors extrait
    dans ce répertoire.


    REQF[] est identique, mais il demande un fichier.

    Exemple :
        c:dir [f] > REQF[Choisissez un fichier]


    REQV[] demande a l'utilisateur de choisir un volume.

    REQT[] demande a l'utilisateur de rentrer un texte, c'est utile
    pour entrer des arguments, par exemple :

        c:cpu REQT[entrer les arguments de cpu]


=========================================================================

Commande SURE

    La commande SURE[texte] affiche une requêtre avec le texte [texte]
    et avec deux boutons : oui / non.
    Si l'utilisateur choisi non, la ligne de commande est annulée.
```

Si l'utilisateur choisi oui, ClassAction exécutera la ligne de commande sur la partie DROITE de la commande Sure.

Exemple :
SURE[Détruisons nous ce fichier ?]C:delete [f]

Ceci affichera une requête demandant à l'utilisateur de répondre oui ou non à la question : "Détruisons nous ce fichier ?".
Si l'utilisateur répond non, rien ne se passe ; si l'utilisateur répond oui, alors C:delete [f] est exécuté.

==========

Bien sûr, vous pouvez combiner autant d'arguments / commandes dans une ligne de commande, que vous le voulez.

Exemple :
SURE[Je le renomme ?]c:rename [f] REQF[Donne moi un nouveau nom]

1.17 Améliorations futures

Voici ce que j'aimerais rajouter à ce programme dans les futures versions :

- Concept de Famille de Classes :
  Par exemple, la famille Images contient GIF, IFF, TARGA, JPEG... et la possibilité de choisir un filtre selon la famille dans la requête de répertoires. Grâce à cela, il sera possible de filtrer les images, sons...
- Et bien sûr, tout ce que vous me demanderez :)

1.18 Remerciements

Je voudrais remercier les personnes suivantes :

- Mireille (pour sa patience...)
- Philippe Thomas (pour des suggestions, aides, beta-tests, le guide en Français et UTT utilisé lors de l'install)

  Hey Phil, si tu lis ça, clique içi
- Bruno Durremberger (pour des beta-tests)
- Jean-Michel Dessolas (pour des beta-tests sur un A4000/40)
- Obvious Implementations Corp (pour Dice C Pro)
- Nico Francois pour la ReqTools library

- Georg Hörmann pour la *FANTASTIQUE* xfdmaster library
- Stefan Stuntz pour la superbe Magic User Interface .
- Tous les personnes qui m'ont contactes pour me reporter des bugs ou pour me faire des suggestions.
- Tous les utilisateurs enregistrés

1.19 L'auteur

Vous pouvez me contacter à l'adresse suivante :

Gasmi Salim
6, rue des Hirondelles

67380 Lingolsheim
France

Web: http://www.sdv.fr/pages/salim

E-Mail: salim@sdv.fr

1.20 Utiliser ClassAction

L'utilisation de ClassAction est VRAIMENT simple.

Sélectionner simplement un fichier dans le requester correspondant.
Vous pouvez aller sur le répertoire précédent grâce au bouton 'Parent'

Après avoir sélectionné un fichier, apparaitra dans la liste de droite,
la classe du fichier et les actions correspondantes.
Sélectionnez simplement l'action souhaitée...

Si vous double-cliquez sur un fichier, la première action définie sera
automatiquement lancée.

ClassAction est aussi une AppWindow c'est a dire que vous pouvez jeter
une icone sur sa fenetre principale .

Vous pouvez invoquer le programme de preference en cliquant sur
le boutton 'Prefs' .

Pour quitter, utiliser le bouton 'Fin'.

Pour réduire la fenêtre en une AppIcone, il suffit de la fermer.
Double cliquez sur l'AppIcone pour réouvrir la fenêtre.

Une fois ClassAction transformé en AppIcone, il suffit de jeter des
icones dessus pour que le type du fichier soit reconnu.
Si la classe dispose d'une seule action possible, alors celle-ci est
automatiquement exécutée.
Sinon, une fenêtre propose la liste des actions, il ne reste plus qu'à
en choisir une.

Pour utiliser le gestionnaire de fichiers, c'est tout aussi simple
il suffit de selectionner l'onglet "gestion" pour voir apparraitre
une seconde listview avec les bouton "copier", "renommer" etc etc.

Il vous faut juste savoir que la liste de GAUCHE est TOUJOURS
la source et que celle de DROITE TOUJOURS la destination .
Aussi que TOUTES les actions du gestionnaire de fichiers s'operent
sur la liste de GAUCHE (source).

Vous pouvez echanger les chemins entre la source et la destination
avec le bouton <--> .

Les fichiers .infos ne sont pas implicitemment geres par le gestionnaire.

Considerez ce gestionnaire de fichier comme un bonus et non pas
comme un reel gestionnaire.

C'est tout, facile non ???

1.21 Informations Techniques

ClassAction est écrit à 100 % avec DICE C 3.0

Informations diverses :

Le fichier de configuration est un fichier ASCII appelé :
ENVARC:ClassAction/ClassAction.prefs
Les actions génériques sont sauvées dans ENVARC:ClassAction/ClassAction_Gen. ↩
prefs
Les action Tiroir seront sauvées dans ENVARC:ClassAction/ClassAction_Dir.prefs

ClassAction crée un exécutable nommé ClassAction_RunTask, stocké
dans T:
Cet exécutable est utilisé pour lancer des tâches depuis le WB.

Informations sur les bibliothèques :

Bibliothèques ROM utilisées :

exec.library	V37+
dos.library	V37+
intuition.library	V37+
graphics.library	V37+

```
    gadtools.library        V39+
    workbench.library       V37+
    utility.library         V39+


Bibliothèques DISK nécessaires :

    rexxsyslib.library      V39+
    commodities.library     V37+
    asl.library             V39+
    icon.library            V37+
    reqtools.library        V38+
    MUI 3.0+

Bibliothèques DISK utilisées (si disponibles) :

    locale.library          V38+
    datatypes.library       V39+
    xfdmaster.library       V30+



Comment ClassAction détermine une classe :

    1- test si le nom du fichier correspond au motif défini pour la
       classe
    2- test si les offsets définis dans la classe correspondent au
       fichier
    3- décompression du fichier grâce à la xfdmaster.library
    4- test si les offsets définis dans la classe correspondent au
       fichier décompressé
    5- test si le fichier est ASCII (si la commande d'Offset ASCII
       est utilisée)

    Si tout cela échoue, le fichier est déclaré comme 'Type inconnu'.
```

1.22 Installation

```
Pour installer ce logiciel :

UTILISEZ le script standard d'installation livré avec l'archive.

Pour ce faire cliquez sur l'icone installe .


Simple non ?
```

1.23 Enregistrement

```
Si vous lisez ces lignes, c'est que vous voudriez être un utilisateur
enregistré de ClassAction.
```

Laissez moi vous expliquer pourquoi vous devriez vous enregistrer :

Tout d'abord, pour soutenir la meuilleure machine jamais créee,
parce que le futur de l'Amiga dépend des logiciels futurs.
Lorsque vous soutenez un programmeur dans son travail, vous soutenez
votre ordinateur et son futur !!!!

Aussi parce que je passe tout mon temps libre à faire des sharewares.
Si vous les utilisez, pourquoi ne pas envoyer la somme demandée ?
Cela me fera continuer à faire des sharewares.

Mais la version ShareWare est utilisable à 100 %, rien n'a été
enlevé !!! Seulement quelques requesters "ennuyeux" en plus.
Donc si vous l'appréciez vraiment, ENREGISTREZ VOUS !!

La somme demandée pour l'enregistrement est (en fonction de votre
monnaie locale) :

        10 US$ ou 20 DM ou 50 FF.

Envoyez la somme à mon adresse ,
Et vous recevrez la dernière version enregistrée de ClassAction.

N'oubliez pas de mentionner votre adresse complète (y compris le pays).

        Merçi d'avance pour votre support !

                        Sincèrement,

                                Salim

1.24 Licence

Copyright

ClassAction et ClassActionPrefs sont Copyright © 1994-1996
par Gasmi Salim.

ClassAction est un programme shareware. Le package ne doit pas être
altéré de quelque façon que ce soit et ne peut pas être utilisé à des
fins commerciales sans la permission écrite préalable de l'auteur.
Ce message de copyright doit être préservé.

Garantie

Aucune responsabilité ne sera acceptée pour des dommages qui pourraient
résulter de l'utilisation de ce programme. Toute utilisation est à vos
risques et périls. Le logiciel est fourni "tel quel" sans aucune
garantie implicite. La documentation est censée être correcte, mais
l'auteur se réserve le droit de mettre à jour le programme et/ou la

documentation sans en notifier l'utilisateur.

1.25 Historique

```
              Historique de ClassAction V3.1 (c) Salim Gasmi

09/05/96 : V3.1

    - ClassAction possede un petit gestionnaire de fichiers integre.

    - Quelques bugs corriges.


12/03/96 : V3.0 (Mise a jour Majeure)

    - ClassAction et ClassActionPrefs sont maintenant en MUI 3.0+

    - ClassAction peut maintenant avoir jusqu'a 50 boutons de chemins.

    - ClassAction reconnait les repertoires.

    - ClassAction gere la multi-selection.

    - ClassAction a une appwindow.

    - Possibilité de regler l'efficacité de l'aprentissage.

    - Tooltypes CAPREFS et GTLIKE rajoutés.

    - Tooltypes HEIGHT,APPSTART,WINX,WINY,ICONX,ICONY,CX_PRIORITY,CX_HOTKEY
                REQBUG,STARTDIR,WBFONT,PUBSCREEN enlevés et obsoletes.


25/09/95 : V2.8

    - ClassAction dispose maintenant d'une gestion TOTALE de sa
      commoditee et propose une HotKey pour Cacher/Monter l'interface.

    - Le Tooltype CX_HOTKEY a ete rajoute pour gerer la Hotkey

    - Le Tooltype APPSTART peut etre maintenant mis a HIDE
      (APPSTART=HIDE) si on veut que ClassAction demarre caché.

    - Le Tooltype PUBSCREEN a ete rajoute pour permettre l'utilisation
      des ecran publics.

    - Le selecteur de fichiers interne de ClassAction ne liberait
      pas toute la memoire allouée, c'est maintenant corrigé .


11/09/95 : V2.75

    - Les commandes REQT,REQV et REQF n'etaient pas incompatibles
      avec le mode AppIcone mais avec certains programmes comme
```

MagicMenu principalement.

Enormement d'utilisateurs se sont plaints du changement en REQT meme si ils n'utilisaient pas un programme creeant le probleme.

J'ai maintenant rajoute un tooltype (REQBUG) pour laisser le choix a l'utilisateur d'utiliser les REQs ou pas a partir de l'AppIcone.

03/09/95 : V2.7

- Commandes [b],[x],[B],[X],[F],[S] rajoutees
- la commande Arexx ACTION a ete rajoutee
- Bug corrige quand on jetait plusieurs icones sur l'appicone
- Quelques bugs dans les routines de 'render' on etes supprimes
- Les commandes REQD,REQF et REQV sont maintenant changees en REQT quand elles sont utilisees a partir de l'appicone ou d'Arexx. Ces commandes ne sont ompatibles qu'avec le mode fenetre.

28/08/95 : V2.6

- La fenetre de ClassAction est maintenant redimensionnable.
- ClassAction et ClassActionPrefs utilisent la ReqTools.library.
- Ajout de la commande REQT pour choisir un texte.
- Ajout de la commande REQV pour choisir un Volume.
- Ajout des tooltypes WINX et WINY.
- Le requester de la commande 'Apprendre' a maintenant un bouton ALL.
- Le nom du fichier selectionné apparait dans tous les requesters REQ.
- ClassAction utilise maintenant un 'Key File' pour les versions enregistrées qui se place dans S: .

17/07/95 : V2.5

- ClassAction et ClassActionPrefs sont maintenant localisés et un catalogue français est fourni dans l'archive.
- Ajout de la fonction Apprendre dans ClassActionPrefs.
- Pour choisir la première action d'un fichier, il faut maintenant double-cliquer dessus au lieu de le resélectionner.
- Les Actions Génériques sont maintenant synchro et relisent le répertoire courant.

- Le ToolType AUTOSELECT n'est plus utilisé.

- Les requêtes REQ s'ouvrent maintenant dans le répertoire courant.

- Quelques améliorations mineures.

12/06/95 : V2.1

- Ajout de la classe interne 'Actions Génériques'.

- Ajout de la commande exec SURE[].

- Ajout de la commande d'offset ASCII[] pour reconnaitre les fichiers ASCII.

- Ajout des commandes ARexx : AppIconify, Show, Status, GetClass.

- Changement du système de lancement des tâches : maintenant j'utilise Systemtags(). Il n'y a plus besoin de fichiers temporaires.

- Vous pouvez maintenant définir un délai pour le mode d'exécution CLI.

- Ajout de 'string' et "string" pour la définition des offsets.

- Le ToolType OUTPUT est devenu obsolète et n'est plus utilisé, on utilise le nouveau ToolType CLISIZE à la place.

- Le requester des actions est maintenant correctement dimensionné, apparait sous la souris et utilise l'écran public le plus en avant.

- Echange des boutons 'Utiliser' et 'Sauver" et ajout du bouton 'Cancel' dans ClassActionPrefs, pour se conformer au look standard des Prefs. Transformation du bouton 'About' en '?'.

- Les routines de gestion des Cycles Gadgets de ClassActionPrefs n'étaient pas conformes à 100 % et des patchs comme Cycle2Menu faisaient bugger ClassActionPrefs ; c'est enfin corrigé.

- Amélioration du code du requester et la routine d'information : ils sont environ 400 % plus rapides.

- Erreur de la couleur de surbrillance dans la Listview corrigée.

- ClassAction ne bloque plus l'écran Workbench lorsqu'il est AppIconnifié

- Un méchant bug a été trouvé et supprimé dans ClassActionPrefs.

- Le bouton de droite ne montre les Assigns que lorsque la souris se trouve dans le requester et recliquer sur le bouton de droite revient au répertoire courant.

- ClassAction mémorise la position de la fenêtre.

23/05/95 : V2.00 (Mise à jour majeure)

- ClassAction dispose maintenant d'une AppIcone.
- ClassAction est maintenant une commodité.
- ClassAction dispose d'un port ARexx.
- ClassAction utilise les fontes par défaut du WB.
- Ajout du mode exec 'Arexx'.
- Des couleurs différentes pour les répertoires/fichiers.
- Ajout de gadgets Haut/Bas pour les actions dans ClassActionPrefs.
- Ajout du gadget 'Utiliser' dans ClassActionPrefs.
- Les classes sont maintenant triées dans la liste de ClassActionPrefs.
- Ajout des ToolTypes : APPSTART, ICONNAME, ICONX, ICONY, CX_PRIORITY, WBFONT, OUTPUT, ICONFILE.
- Lorsque la fenêtre est zoomée, elle a maintenant la bonne hauteur.
- Nouveau format de sauvegarde (CASF20).
- Suppression du bouton Suffix/Prefix, remplacé par le gadget MatchName qui accepte les wildcards (motifs).
- Déplacement du fichier de config dans ENVARC:
- A script installer est maintenant fourni avec l'archive.
- Optimisations du code.

05/05/95 : V1.43

- Nettoyage du code du requester, il est environ 5% plus rapide.
- ClassAction recherche son nom dans la structure WBstartup et peut donc être renommé.

02/05/95 : V1.42

- Ajout du mode exec 'No Cli' dans ClassActionPrefs.
- Des "" sont toujours ajoutés autour des noms de fichiers, même si cela n'est pas nécessaire.
  C'est plus simple pour les scripts ARexx.

06/04/95 : V1.4

- Ajout du ToolType HEIGHT.

- Optimisation du code optimization.

22/02/95 : V 1.31

- Si vous cliquez deux fois sur le même fichier, la première action sera lancée (c'est plus facile que de sélectionner l'action à la main).

- Cette version est à présent ShareWare et vous devez vous enregistrer afin de recevoir la version enregistrée.

15/01/95 : V 1.3

- Ajout des commandes interactives REQD[] et REQF[].

- Ajout du gadget 'Copy' dans ClassActionPrefs.

- Quelques améliorations mineures.

- Les Beta-testeurs indiquent que cette version est vraiment stable.

25/11/94 : V 1.22

- Première Version Publique.

- Ajout du bouton 'Info'.

- ClassAction vérouillait les répertoires qu'il lisait, sans les dévérouiller ensuite. Corrigé.

- Optimisation du code.

- Quelques autres bugs mineurs corrigés.

08/11/94 : V 1.21

- Le programme se plantait avec des unités de disque vides... Corrigé.

- Bug corrigé avec Volume/Name.

- La ligne <..> n'apparait plus lorsqu'on est sur la racine d'un volume.

07/11/94 : V 1.2

- Nouvelle interface, j'ai inclu mon propre et rapide requester de fichiers.

- Ajout des ToolTypes STARTDIR et DRIVE1 à DRIVE11.

01/11/94 : V 1.1

- Utilise maintenant la xfdmaster.library pour reconnaitre et décompresser les fichiers.

- Utilisation des ToolTypes pour la configuration (DECRUNCH,AUTOSELECT).

- 'Unknown Class' est maintenant une classe interne avec un nombre illimité d'actions.

- Nouveau format de sauvegarde (CASF11).

16/10/94 : V 1.0

- Nouveau format de sauvegarde (CASF10).

- La fenêtre a maintenant un gadget de Zoom.

- Ajout de nombreuses définitions de classes.

10/10/94 : Beta Version

- Correction du bug de fichier temporaire.

- Correction d'une erreur dans l'incrémentation de l'Offset.

- Utilisation de l'asl.library pour le file requester.

01/10/94 : Alpha Version

1.26 Les commandes ARexx

ClassAction dispose d'un port ARexx appelé : ClassAction.01

Voici une liste de toutes les commandes ARexx :

==========

Quit

Ne fait que quitter ClassAction...

==========

Use

Force ClassAction à recharger le fichier de config.

==========

Ver

```
    Retourne la version de ClassAction.

=====================================================================

Status

    Retourne l'état actuel de ClassAction.

    Retourne 0 : ClassAction est AppIconnifié.
    Retourne 1 : ClassAction est en mode fenêtre.

=====================================================================

AppIconify

    Force ClassAction à s'AppIconifier.

    Si ClassAction est déjà en AppIcone, cette commande ne fait
    rien.

=====================================================================

Show

    Force ClassAction à afficher la fenêtre principale.

    Si ClassAction est déjà en fenêtre, cette commande ne fait
    rien.

=====================================================================

Load <nom de fichier>

    ClassAction essaie de charger le fichier <nom de fichier>, et
    affiche le requester des actions, pour que l'utilisateur puisse
    choisir l'une d'entre elles.

    Si le fichier n'existe pas, cette commande ne fait rien.

=====================================================================

GetClass <nom de fichier>

    ClassAction essaie de charger le fichier <nom de fichier> et
    retourne la classe de ce fichier.

    Si le fichier n'existe pas, cette commande ne fait rien.


=====================================================================


Action <nom de fichier> <Action Generique>

ClassAction va essayer de lancer la premiere action qui repondra
a <action generique> sur le fichier <nom de fichier>
```

un exemple :

Action ram:toto.lha extr

lancera la premiere action dont le nom contient extr sur le fichier
ram:toto.lha

la formule generique est pratique , car vous n'avez pas a donner
le nom complet de l'action a utiliser , un bout de son nom suffit.

1.27 Après l'installation

Après l'installation, vous devriez avoir une ClassAction en état de
marche, avec un grand nombre de classes prédéfinies, mais avec très
peut d'actions (la plupart des classe n'ont même pas d'action définie).

C'est à vous de configurer les actions en fonction de votre système
et de vos préférences.

Pour cela, chargez simplement ClassActionPrefs et configurez le à votre
goût...

Si vous ne savez pas comment configurer les actions et les classes,
lisez ce guide :)).

Après avoir configuré tout cela, essayez encore de configurer le
comportement de ClassAction avec ses ToolTypes.

Cela peut prendre du temps avant de parvenir à une 'chouette' config.
Mais une fois que s'est fait, c'est vraiment SUPER !!!

Bien, maintenant lisez ce guide et Bonne Chance.

1.28 Message pour Philippe THOMAS

Salut Philippe !!!

Je voulais simplement te remercier pour toute l'aide que
tu m'as apporté à la création de ce programme.

Quasiment toutes les améliorations de la version 2.0, c'est
toi qui me les a proposées, parfois même avec insistance,
style le resize de la fenêtre que je n'ai toujours pas fait.. :(

Encore merçi pour tous les appels téléphoniques ; parfois plus
d'une heure à discuter des Hooks, SystemTagList, de bugs etc...
et ce, même en periode d'exams.

Franchement si ClassAction commence à être cool, c'est beaucoup
grâce à toi, je crois que j'aurais eu la flemme de le paufinner

autant si tu n'étais pas là.

Bref, ce programme est aussi un peu le tien.

Okay Phil, à la prochaine.

Salim.

PS : Non, non ton processeur il est bien, il est pas buggé... :)).

1.29 Bugs referencés

ClassActionPrefs ne doit pas etre lance plusieurs fois simultanement
car sinon le 2nd ClassActionPrefs lance , crashes ....

Je cherche le pourquoi de la chose .. en attendant ne lancez pas 2 CAP
en meme temps , de toutes facon cela ne sert a rien :)))