

ClassAction

Gasmi Salim

COLLABORATORS

	<i>TITLE :</i> ClassAction		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Gasmi Salim	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ClassAction	1
1.1	ClassAction 2.8 Guide	1
1.2	Was ist ClassAction?	2
1.3	Mehr über ClassAction	3
1.4	System-Vorraussetzungen	3
1.5	Konfiguration	3
1.6	ClassAction Prefs	8
1.7	Benutzen der Lernen-Funktion	8
1.8	Was ist eine Klasse	9
1.9	Definieren einer neuen Klasse	10
1.10	Definieren einer neuen Aktion	13
1.11	CLI-Modus	14
1.12	WB-Modus	14
1.13	NO CLI-Modus	15
1.14	ARexx-Modus	15
1.15	Argumente	15
1.16	Kommandos	16
1.17	Zukünftige Erweiterungen	18
1.18	Über	18
1.19	Der Autor	19
1.20	Benutzung von ClassAction	19
1.21	Technische Informationen	19
1.22	Installation	21
1.23	Registrierung	21
1.24	Rechtliches	22
1.25	Geschichte	23
1.26	Die Arexx-Kommandos	28
1.27	Was sollten Sie nach der Installation machen	30
1.28	Message for Philippe THOMAS	30
1.29	Bekannte Fehler	31

Chapter 1

ClassAction

1.1 ClassAction 2.8 Guide

```
ClassAction
Version 2.8

~Was~ist~ClassAction~~~~~
    Lesen Sie dies bitte zuerst

~System~Voraussetzungen~~~
    Was Sie brauchen

~Installation~~~~~
    Wie ClassAction installiert wird

~Installation,~Teil~2~~~~~
    Installation Part II

~Benutzung~von~ClassAction
    Wie benutze ich dieses Programm

~Konfiguration~~~~~
    ToolTypes ändern

~ClassAction~Prefs~~~~~
    'Klassen' und 'Aktionen' einstellen

~Die~Arexx~Kommandos~~~~~
    ARexx-Port und -Kommandos

~Zukünftige~Erweiterungen~
    Was wird sich ändern

~Geschichte~&~Entwicklung~
    Von Anfang an...
```

~Registrierung~~~~~
 Registrierung - warum und wie

~Licence~

~Der~Autor~

~Danksagungen~

~Bekannte~Fehler~

1.2 Was ist ClassAction?

ClassAction ist ein kleines Hilfsmittel, um das Leben aller Festplattenbesitzer zu erleichtern.

Wenn Sie eine Festplatte besitzen, dann haben Sie auch eine ganze Anzahl von verschiedenen Files: Programme, Module, Bilder, Quelltexte, Animationen, Töne...

ClassAction bestimmt für Sie die Art des ausgewählten Files und zeigt Ihnen eine Liste aller Aktionen die Sie damit ausführen können.

Zum Beispiel, wenn Sie ein GIF-Bild auswählen, wird es von ClassAction als ein GIF_Klasse-Bild erkannt und es wird eine Liste mit Aktionen aufgezeigt, wie 'anzeigen' oder 'bearbeiten'.

ClassAction ist höchst konfigurierbar, Sie können Ihre eigenen Klassen und Aktionen selbst bestimmen.
Aktionen benutzen externe Programme, dadurch können Sie z. B. Ihren bevorzugten Gif-Anzeiger verwenden, um sich Ihre Gif-Bilder anzuschauen (oder etwas ähnliches).

ClassAction hat ein AppIcon, einen Arexx-Port, ist lokalisiert und ist ein Commodity.

ClassAction benutzt die xfdmaster.library zum automatischen entcrunchen gecrunchter Files.
Mit dieser Möglichkeit kann sogar eine crunched-Klasse festgelegt werden.

ClassAction benötigt sehr wenig Speicher und CPU-Zeit.

Wenn Sie sich Zeit nehmen um ClassAction zu konfigurieren, können Sie alles damit machen!!! Es ist eine einfache Art um mit Files umzugehen.

Bevor Sie sich entscheiden, dieses Programm zu löschen - probieren Sie es doch einfach mal aus!!!

Wenn Sie mehr über ClassAction wissen wollen, klicken Sie auf
~mehr~

1.3 Mehr über ClassAction

ClassAction und ClassActionPrefs sind (C) 1994-95 bei Gasmi ↔
Salim

Diese Packet wurde als ShareWare herausgegeben. Sie können es gerne benutzen!!! und weitergeben, solange Sie nichts verändern. ABER, wenn Sie es regelmäßig benutzen, müssen Sie sich registrieren lassen!!!

PD-Distributoren ist es erlaubt, das ClassAction-Paket in Ihre Sammlung aufzunehmen - solange Sie mich wissen lassen, daß Sie es aufgenommen haben.

Dies ist Release 2.8 von ClassAction.
Wenn Sie es mögen und benutzen, MÜSSEN Sie sich
~registrieren~
lassen.

Ich hoffe, daß Sie dieses Programm nützlich finden.
(Na ja, ich finde wenigstens nützlich :))

Für technische Einzelheiten über ClassAction klicken Sie auf

~Technische~Informationen~
.

1.4 System-Vorraussetzungen

Um ClassAction verwenden zu können, benötigen Sie folgendes:

- o Amiga OS 3.0 oder höher
(Vielleicht erscheint eine OS 2.0 Version)
- o eine Festplatte
(ClassAction ist ohne Festplatte nutzlos)

Das war's schon!!!

1.5 Konfiguration

ClassActions Oberfläche und Möglichkeiten sind über die ToolTypes zu konfigurieren.

Um einen ToolType-Eintrag zu ändern, klicken Sie das ClassAction-Icon an und wählen Sie aus dem Workbench-Menü den Eintrag "Piktogramm/Information ...".

Dies ist die Liste aller ToolTypes:

Anmerkung: Wenn ein benötigtes ToolType nicht gefunden wird, dann wird der jeweilige Default-Wert verwendet.

DONOTWAIT

Löschen Sie dieses ToolType bitte NICHT, es wird benötigt, wenn ClassAction aus der WBStartup-Schublade gestartet wird.

CX_PRIORITY

Dies ist die Commodity-Priorität von ClassAction.

Default-Wert ist: 0

CX_HOTKEY

Dies ist der Hotkey um ClassAction zu öffnen/verstecken.

Wenn Sie ihn benutzen, während ClassAction versteckt oder als AppIcon zu sehen ist, wird das Programmfenster geöffnet. Wenn Sie ihn benutzen, während das Programmfenster geöffnet ist, dann wird es versteckt.

Default-Wert ist: LALT C (LinkeAlt + Shift + c)

REQBUG

Wenn Sie Programme wie MagicMenu (oder andere) benutzen, während sich ClassAction im AppIcon-Status befindet, kann es zu Störungen mit REQ-Kommandos kommen. Diese können zum Absturz des Rechners führen...

Wenn Sie dieses ToolType auf YES (z. B.: REQBUG=YES) setzen, ändert ClassAction die REQV-, REQF- und REQD-Kommandos in ein REQT-Kommando um, falls die Aktion über das AppIcon gestartet wird.

Default-Wert ist: NO

APPSTART

Setzen Sie dieses ToolType auf YES,NO oder HIDE

NO : ClassAction startet mit einem Fenster

YES : ClassAction startet als AppIcon
HIDE : ClassAction startet versteckt

Default-Wert ist: NO

=====

STARTDIR

Dieses ToolType legt das Verzeichnis fest, das Sie angezeigt bekommen möchten, wenn Sie ClassAction zum ersten Mal starten.

Default-Wert ist: Current directory (das aktuelle Verzeichnis)

=====

WBFONT

Setzen Sie diese ToolType auf YES (z. B.: WBFONT=YES), wenn Sie möchten, daß ClassAction den eingestellten Workbench-Font verwendet. Wenn Sie es auf YES setzen, muß die Fontgröße kleiner als 15 sein, sonst verwendet ClassAction den topaz 8-Font.

Wenn Sie das ToolType auf NO setzen, benutzt ClassAction den topaz 8-Font.

Default-Wert ist: YES

=====

DECRUNCH

Wenn Sie dieses ToolType auf YES setzen, versucht ClassAction gecrunchte Files mittels der xfdmaster.library zu entpacken.

Es ist notwendig, wenn Sie gecrunchte Files erkennen lassen wollen.

Wenn Sie dies tatsächlich so tun wollen, werden die Files vorab in den Speicher geladen und es wird dem entpackten File ein Puffer zugewiesen.

Dies belegt Speicherplatz (ungefähr 2,5 x soviel wie das File). Falls Sie also nicht genügend Speicher haben, dann sollten Sie dieses ToolType auf NO setzen.

Default-Wert ist: NO

=====

HEIGHT

Dieses ToolType legt die Höhe des ClassAction-Fensters fest.

Die minimale Höhe hängt von Ihrem WB-Font ab. Wenn Sie eine zu kleine Höhe einstellen, verwendet ClassAction die kleinste mögliche Höhe.

Default-Wert ist: 0 (verwendet die kleinste mögliche Fensterhöhe)

WINX and WINY

Dieses ToolType legt die Standard-Koordinaten des Fensters fest.

Default-Werte sind: 10

ICONX and ICONY

Dieses ToolType legt die Koordinaten des AppIcons fest.

Beachten Sie bitte, daß es sich hierbei nur um einen Vorschlag handelt. Wenn die ausgewählten Koordinaten durch ein anderes Icon belegt sind, werden sie von der Workbench ignoriert und das AppIcon so nahe wie möglich daneben plazierte.

Default-Wert ist: No icon position (keine Icon-Position)

PUBSCREEN

Wenn Sie dieses ToolType auf YES (PUBSCREEN=YES) setzen, öffnet ClassAction sein Fenster auf dem vordersten Public-Screen.

Default Value is : NO

ICONNAME

Dieses ToolType legt die Beschriftung des AppIcons fest.

Wenn Sie z. B. ICONNAME=Drop eintragen, erscheint unter dem AppIcon 'Drop'.

Diese Feld kann auch leer bleiben (z. B.: ICONNAME=), dann erscheint keine Beschriftung unter dem AppIcon

Default-Wert ist: ClassAction

ICONFILE

Dieses ToolType legt das AppIcon-File fest.

Dadurch können Sie Ihr eigenes AppIcon für ClassAction verwenden. Wenn Sie dieses ToolType auf ein Icon-File setzen, benutzt es ClassAction als sein AppIcon-Bild.

Anm.: Verwenden Sie keine .info-Files als Filenamen.

Beispiel: ICONFILE=Sys:icons/Kopf

verwendet das Kopf.info-Icon aus dem Verzeichnis
Sys:Icons/ als AppIcon.

Falls ClassAction Ihr Icon-File nicht laden kann, verwendet es
sein eigenes Default-Icon (ClassAction.info).

Default-Wert ist: "" (verwendet das Default-AppIcon)

=====

CLISIZE

Dies ist die Größe des CLI-Ausgabefensters, wenn SystemTags()
verwendet werden.

Die Syntax ist: DEVICE:TopX/TopY/Width/Height/Title

Dadurch können Sie ein anderes Device als CON: für Ihre
CLI-Fenster verwenden, oder eine neue Fensterposition und
-größe wählen.

WARNUNG:

Wenn Sie dieses ToolType auf einen übertriebenen Wert einstellen,
wird ClassAction das CLI nicht starten.
Verändern Sie es NICHT, wenn Sie nicht genau wissen, was Sie da tun.

** NIEMALS ** Device-Kommandos wie AUTO, CLOSE, WAIT hier eintragen.
Dies erledigt ClassAction für Sie.

** NIEMALS ** Leerzeichen in diese Zeile einfügen.

In der Version 2.0 war ein ähnliches ToolType OUTPUT vorhanden, das
aber nicht länger gebraucht und benutzt wird.

Default-Wert ist:
CON:0/0/640/100/ClassAction_Output_Window

=====

DRIVE1 bis DRIVE11

Setzen Sie diese ToolTypes auf einen gültigen Pfad, dann erscheinen
sie in den Knöpfen des ClassAction-Fensters.
Dies ist nützlich, wenn Sie schnell ein Verzeichnis auswählen wollen.

Die Syntax ist:

DRIVEx=<Knopf-Text>,<Pfad>

Z. B.: DRIVE9=Jpeg,dh0:gfx/pictures/jpeg

Dadurch erscheint als Beschriftung des Knopfes Nr.9 'Jpeg' und der
Pfad ist auf dh0:gfx/pictures/jpeg gesetzt.

Sie können natürlich genauso den <Knopf-Text> ohne den <Pfad> an-

geben, z. B.: DRIVE3=dh0:libs.

In diesem Falle wird der Text des Knopfes auch als Text des Pfades verwendet. Aber benutzen Sie keine zu langen Texte, weil diese in den schmalen Knöpfen nicht angezeigt werden können.

Default-Wert ist: "" (keiner)

1.6 ClassAction Prefs

Klassen und Aktionen zu definieren, ist die Hauptverwendung dieses Programmes. Sie müssen dazu das ClassActionPrefs-Programm benutzen.

ClassActionPrefs zu benutzen sollte einfach sein... auf geht's!!

Das Fenster ist in zwei Teile aufgegliedert: Klassen und Aktionen.

Zuerst wählen Sie eine Klasse, die dazugehörige Aktion wird im Aktions-Teil aufgezeigt.

Um eine Klasse hinzuzufügen oder zu löschen, klicken Sie einfach auf den dazugehörigen Knopf.

Für die Aktionen gilt das gleiche Konzept...

- ~1.~Was~ist~eine~Klasse~~~~~
- ~2.~Anlegen~einer~neuen~Klasse~~~~
- ~3.~Definieren~einer~neuen~Aktion~
- ~4.~Die~Lernen~Funktion~~~~~

1.7 Benutzen der Lernen-Funktion

Die Lernen-Funktion wird mitgeliefert, um Ihnen dabei zu helfen, neue Klassen zu definieren.

Wenn Sie neue Klassen festlegen, müssen Sie Offsets (Verzweigungen, weiterführende Beschreibungen) dafür definieren. Und es mag mühselig und langweilig sein, um abzuschätzen, welche Offsets die neue Klasse treffend definieren könnten.

Genau das ist der Grund, warum diese wunderbare Lernen-Funktion geschaffen wurde!

Um sie zu benutzen, müssen Sie folgendermaßen vorgehen:

1 : definieren Sie wie üblich Ihre Klasse, indem Sie Name und

Klassen-Name eintragen

- 2 : klicken Sie auf den 'Lernen'-Knopf
- 3 : wählen Sie aus dem dann erscheinenden Requester so viele Files aus, wie Sie wollen - aber nur solche, die auch zu der gleichen Klasse gehören, die Sie festlegen wollen.
Je mehr Files Sie aussuchen, umso besser wird das Ergebnis.
ClassAction versucht nun die Offset-Beschreibungen herauszufinden.
(Um mehrere Files gleichzeitig auszuwählen, verwenden Sie bitte die Shift-Taste.
- 4 : Nach der Analyse erscheint ein Fenster mit den gefundenen Offsets.
- 5 : Die Offsets können nun per Hand geändert werden, falls das notwendig sein sollte.
- 6 : Klicken Sie auf 'Akzeptieren' um die Offsets für Ihre Klasse zu übernehmen.

WARNUNG:

Damit die Lernen-Funktion richtig arbeitet, müssen Sie sicher sein, daß:

- * ALLE AUSGESUCHTEN FILES ZUR GLEICHEN KLASSE GEHÖREN.
- * ALLE AUSGESUCHTEN FILES NICHT GEGRÜNDET SIND.

Das war's schon!!! Einfach, oder?

1.8 Was ist eine Klasse

Eine Klasse ist eine Familie von Files. Beispielsweise C-Files kann man zu eine Klasse zusammenfügen, die man dann C-Klasse nennen kann.

Mit ClassAction können Sie so viele Klassen definieren, wie Sie wollen - so lange Sie erklären können, woran man sie erkennt.

Um zu erklären, wie man eine Klasse erkennt, gibt es zwei Möglichkeiten:
am MatchName (gemeinsamer übereinstimmender Name, Gruppenname) und am Inhalt der Files.

MatchName wird benutzt, wenn man Files über ihren Namen erkennen lassen will.
Offsets hingegen werden verwendet, wenn man Files an ihrem Inhalt erkennen lassen will.

Es gibt zwei programminterne Klassen, die man nicht entfernen kann. Sie werden in der Farbe 'weiß' in der Klassen-Liste dar-

gestellt.

Die erste wird als "neue Klasse" bezeichnet (Sie können diese Bezeichnung allerdings umbenennen.)
Diese Klasse enthält alle Files, die ClassAction nicht erkennen kann.

Die zweite wird als "Allgemeine Aktionen" bezeichnet (dies können Sie nicht ändern) und enthält Aktionen, die in ALLEN anderen Klassen aufgezählt werden...

Anm.: Wenn Sie die Aktion 'Kopieren' für alle Klassen anlegen wollen, dann können Sie dies tun, indem Sie die Aktion für jede einzelne Ihrer Klassen definieren - aber das dauert laaaange und ist nervend.

Der einfachere Weg ist es, wenn Sie diese Aktion in den "Allgemeinen Aktionen" anlegen, dadurch wird 'Kopieren' für alle Klassen angezeigt.

Die allgemeinen Aktionen werden in weiß in dem ClassAction-Aktionsfenster angezeigt und sind nur im ClassAction-Fenster sichtbar. Sie werden nicht angezeigt, wenn sich ClassAction im AppIcon-Status befindet oder über ein

Arexx-Kommando
gestartet wird.

1.9 Definieren einer neuen Klasse

Klicken Sie nur auf den 'dazu'-Knopf auf der Klassen-Seite um eine Klasse hinzuzufügen.

Eine Klasse hat 3 Eigenschaften:

- einen Namen
 - den MatchName (Gruppennamen)
 - Offsets (Inhalte)
- o Der Name ist einfach der Klassenname, den Sie selbst wählen können.
WARNUNG : ein Klassenname darf nur einmal vergeben werden.
 - o Der MatchName ist jeder gewöhnliche AmigaDOS-Ausdruck, wie:

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

(Lesen Sie dazu auch die AmigaDOS-Beschreibung aller wildcards (Platzhalter)).
WARNUNG : Benutzen Sie nicht das Wildcard '*', sondern verwenden ersatzweise '#?' dafür.

Der MatchName ist nicht casesensitiv, Sie können also z. B. 'toto.C' auch als '#?.c' beschreiben.
-

Wenn Sie eine Klasse mit einem MatchName beschreiben, sollten Sie sich sicher sein, daß die Beschreibung richtig und gut ist.

Beispiel: Wenn Sie mit dem MatchName '#?.gif' die GIF-Klasse festlegen, werden alle Files mit der Endung '.gif' als GIF-Files eingeordnet.

Aber sind Sie sich auch sicher, daß ALLE '.gif-Files' auch GIFs sind, bzw. daß alle Ihre GIFs die Endung '.gif' haben?

Sie sollten deshalb MatchName nur in zwei Situationen verwenden:

- der MatchName ist eindeutig für diese Klasse
(z. B.: #?.info ist gut genug als MatchName für eine Icon-Klasse)
- Sie habe keine andere Wahl
(z. B.: wie sollten Sie einen C-Quelltext anders erkennen als am #?.c)

- o In allen anderen Fällen sollten Sie daher Offsets verwenden.

Ein Offset ist ein Teil eines Files, in dem man etwas finden, das erkannt werden kann.

Z. B. beginnen GIF-Bilder immer mit dem String 'GIF' an Offset 0.

Es gibt drei Formen um Offsets zu definieren:

=====

Syntax #1 : Offset,HexString

Offset ist die DECIMAL-Nummer, die das Offset besitzt.
HexString ist ein HEX-String, der an dem Offset gefunden werden kann.

Beispiel 1 : 0,4f4a bedeutet, daß das Offset mit den Bytes \$4f und \$4a an der Position 0 beginnt.

Beispiel 2 : 9,448b3c bedeutet, an Byte #9 befindet sich \$44 \$8b \$3c.

=====

Syntax #2 : Offset,'String'

Offset ist die DECIMAL-Nummer, die das Offset besitzt.
String ist ein ASCII-String, der an dem Offset gefunden werden kann.

Beispiel 1 : 0,'GIF' bedeutet, daß das File mit dem String

'GIF' beginnen muß.

Beispiel 2 : 9,'Fußar' bedeutet, das sich an Byte #9 der String
'Fußar' befindet.

=====
Syntax #3 : Offset,"String"

Offset ist die DECIMAL-Nummer, die das Offset besitzt.
String ist ein ASCII-String, der an dem Offset gefunden werden
kann.

Bachten Sie den Unterschied zu dem vorherigen Beispiel. Hier
verwenden wir " um den String zu definieren, in Syntax #2
wurde ' verwendet.

Es ist das gleiche Konzept wie in Syntax #2, nur wird hier
die SCHREIBWEISE nicht beachtet.

Als Beispiel: ein Amiga Guide beginnt immer mit dem String:
@database in Klein- oder Großbuchstaben.

Wenn Sie die Methode #2 mit 0,'@database' verwenden um ein
AmigaGuide-File zu erkennen, wird ClassAction ein File, das
mit @DATABASE beginnt, nicht als AmigaGuide-File erkennen.

Das funktioniert nur, wenn Sie mit Syntax #3 folgendes
definieren: 0,"@database".

=====
Sie können bis zu 5 Offsets festlegen um eine Klasse zu definieren.
Ein File wird dann erst dann zu dieser Klasse gezählt, wenn alle 5
Offsets darauf zutreffen.

Beispiel: wenn die Klasse X definiert wird als

Offset #1 : 0,4a8b6c
Offset #2 : 58,14

Alle Files, die mit 4a8b6c beginnen UND \$14 an Byte #50 haben,
werden als X erkannt.

Um verschiedene Offsets zu definieren, klicken Sie nur auf den
Wechselknopf 'Offset #'. Dadurch wird das nächste Offset aktiviert.

Bitte beachten : Die ASCII-Klasse

Es gibt ein eingebautes Offset-Kommando namens : ASCII[]

Wenn Sie dieses Kommando in Offset #1 (z. B. Offset#1=ASCII[])

eintragen, dann stimmt es mit ASCII-Files überein. Aber ClassAction wird dies immer versuchen, wenn irgend etwas fehlgeschlagen ist. Dank dieser Vorgehensweise werden AmigaGuide-Files (die ASCII-Files sind) nicht als ASCII erkannt, wenn Sie bereits eine AmigaGuide-Klasse definiert haben.

Normalerweise werden Sie dies nicht gebrauchen, weil ich im Standart-Prefs-File bereits festgelegt habe, daß die Klasse 'ASCII' mit diesem Kommando definiert wird.

1.10 Definieren einer neuen Aktion

Wenn erst einmal eine Klasse definiert ist, sollten Sie Aktionen dafür festlegen. Jede Klasse kann so viele Aktionen haben, wie Sie möchten.

Klicken Sie einfach auf den 'dazu'-Knopf auf der Aktionen-Seite um eine Aktion hinzuzufügen.

Jede Aktion hat 5 Eigenschaften:

- einen Namen
 - einen 'run mode' (die Betriebssystemebene, auf der die Aktion läuft)
 - eine Stackgröße (nur im 'run mode' Cli)
 - eine Verzögerung (nur im 'run mode' Cli)
 - ein 'exec command' (auszuführendes Kommando)
- o Name ist der Name der Aktion.
 - o Run mode kann sein:
 - ~Cli~
 - ,
 - ~WB~
 - ,
 - ~No~Cli~
 - oder
 - ~ARexx~
 - .
 - o Exec command ist eine gültige AmigaDOS-Kommandozeile und kann Parameter enthalten. SIE SOLLTEN dabei immer den kompletten Pfad der DOS-Befehle angeben.

Beispiel: benutzen Sie C:Copy anstelle von Copy in den Kommandozeilen.

Sie können über die Kommandozeilen

~Argumente~

und

~Kommandos~

einfügen, die in ClassAction enthalten sind.

Knöpfe

- o Der 'U' Up-(Aufwärts-) und 'D' Down-(Abwärts-)Knopf zum Auswählen der Aktionen.
- o The 'Laden'-Knopf läßt Sie auswählen, welche DOS-Befehle Sie in die Kommandozeile einfügen möchten.
- o Der 'Comm'-(Kommando-)Knopf öffnet einen Requester, der alle für eine Kommandozeile möglichen Argumente und Kommandos auflistet.
- o Mit dem Knopf 'Copy' können Sie alle Aktionen einer anderen Klasse in die gerade aktuelle Klasse kopieren. Klicken Sie einfach auf 'Copy' und suchen Sie sich eine Klasse aus, von der Sie die Aktionen kopieren wollen, Das ist nützlich, wenn sich gleiche Aktionen in verschiedenen Klassen befinden.

1.11 CLI-Modus

**** CLI-Modus ****

Wenn Sie 'Cli' gewählt und sich eine Aktion ausgesucht haben, dann wird diese Aktion vom Cli aus gestartet und die Stackgröße des Cli richtet sich nach der System-Vorgabe (Default ist 4096).

Im Cli-Modus wird nur dann ein Clifenster geöffnet, wenn es nötig ist (wenn eine Ausgabe erfolgt).

Sie können die Anzeige-Parameter so festlegen:

Wenn Delay (Verzögerung) negativ ist (z. B. Delay = -1), dann bleibt das Clifenster so lange offen, bis Sie es per Hand mit dem Schließ-Knopf in der linken oberen Ecke schließen.

Wenn Delay auf Null steht (z. B. Delay = 0), dann schließt sich das Cli-Fenster automatisch, sobald der Task sich beendet hat.

Wenn Delay positiv ist (z. B. Delay = n, wobei n größer als 0 sein muß), dann wartet das Cli n-Sekunden bis es sich selbst schließt. Sie können es allerdings auch vorher selbst mit dem Schließ-Knopf beenden.

Die Größe des Clifensters können Sie über das ToolType CLISIZE einstellen (das alte ToolType OUTPUT ist dadurch veraltet).

1.12 WB-Modus

*** WB-Modus ***

Wenn 'WB' ausgewählt wurde, dann öffnet sich kein Clifenster und ClassAction simuliert einen Workbench-Start der Aktion. Das betreffende File wird mit den Werten gestartet, die in seinen ToolTypes eingetragen sind.

WARNUNG: dieser Modus ist nur bei Files möglich, die auch tatsächlich über ein Icon verfügen.

1.13 NO CLI-Modus

*** NO CLI-Modus ***

Wenn Sie 'No Cli' gewählt haben, wird solange kein Cli geöffnet, bis das Programm eine Ausgabe macht. Aber der Task läuft trotzdem weiter über das Cli.

1.14 AREXX-Modus

*** AREXX-Modus ***

Wenn Sie 'AREXX' gewählt haben, wird rx mit den angegebenen Kommandos gestartet. Diese Kommandos MÜSSEN ein Arexx-Skript sein.

Natürlich sollte RexxMaster dafür vorher gestartet sein und es sollte sich das Programm rx im Verzeichnis Sys:rexxc/ befinden.

1.15 Argumente

Zur Zeit sind 8 Argumente möglich:

die ersten 4 Kommandos sind in Kleinbuchstaben: [f] [s] [b] [x] und in Hochzeichen eingeschlossen

[f] : voller Pfad des ausgewählten Files mit Hochzeichen
[s] : voller Pfad des ausgewählten Files ohne Suffix mit Hochzeichen
[b] : Filename des ausgewählten Files mit Hochzeichen
[x] : Filename des ausgewählten Files ohne Suffix mit Hochzeichen

die letzten 4 Kommandos erzielen das gleiche Ergebnis wie die Kommandos vorher, sind aber ohne Hochzeichen

[F] : voller Pfad des ausgewählten Files
[S] : voller Pfad des ausgewählten Files ohne Suffix
[B] : Filename des ausgewählten Files
[X] : Filename des ausgewählten Files ohne Suffix

Beispiel: nehmen wir mal an Sie wollen das File ram:env/sys.prefs

```

        auswählen

[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

```

Beispiele: nehmen wir an, Sie hätten das File ram:main.c ausgewählt

```

* Die Kommandozeile
      c:copy [f] [F].bak
wird dann ersetzt durch
      c:copy "ram:main.c" ram:main.c.bak

* Die Kommandoszeile
      c:copy [f] [S].bak
wird dann ersetzt durch
      c:copy "ram:main.c" ram:main.bak

```

1.16 Kommandos

Zur Zeit sind 5 REQ-Kommandos möglich:

```

REQD[text] : fragt nach einem Verzeichnis (Directory).
REQF[text] : fragt nach einem File.
REQV[text] : fragt nach einem Datenträger (Volume).
REQT[text] : fragt nach einem Text.
SURE[text] : fragt nach, ob der User weitermachen will
              (Sicherheitsabfrage).

```

=====

REQs-Kommandos

```

REQ-Kommandos öffnen eine Reqtools-Requester mit dem
Titel [text].
Dies ist nützlich, wenn Sie interaktive Kommandozeilen
benötigen.

```

Beispiel:

```
bin:lha x [f] to REQD[Wähle ein Verzeichnis zum Entpacken]
```

```

Dies öffnet einen Verzeichnis-Requester, in dem der User ein
Zielverzeichnis wählen kann. Das ausgesuchte File [f] wird
dann in das Zielverzeichnis entpackt.

```

REQF[] ist das gleiche, fragt aber nach einem File.

Beispiel:

```
c:dir [f] > REQF[Wähle ein File]
```

REQV[] fragt den User nach einem Datenträger (Volume).

REQT[] fragt nach einem Text. Das ist nützlich, um beispielsweise nach Argumenten zu fragen:

```
c:cpu REQT[Bitte Argumente für CPU eingeben]
```

WARNUNG:

Bei REQD-, REQF- und REQV-Kommandos kann es möglicherweise im AppIcon-Modus zu Störungen kommen, wenn Sie gleichzeitig Programme wie MagicMenu benutzen

Wenn es im AppIcon-Modus bei Aktionen die die REQD-, REQF- und REQV-Kommandos verwenden, öfter zu Guru-Meldungen kommt, dann versuchen Sie bitte herauszubekommen, mit welchem Programm diese Störungen auftreten. Oder Sie setzen das REQBUG auf YES.

Wenn Sie das TollType auf YES gesetzt haben und die REQD-, REQF- und REQV-Kommandos über das AppIcon benutzt werden, dann werden diese Kommandos automatisch in ein REQT-Kommando umgewandelt (sicheres Kommando).

SURE-Kommando (Sicherheitsabfrage):

Das SURE[Text]-Kommando öffnet einen Requester mit dem Text [Text] und mit 2 Knöpfen: Ja/Nein.
Wenn Sie 'Nein' wählen, dann wird die Kommandozeile abgebrochen.
Wenn Sie 'Ja' wählen, dann führt ClassAction den RECHTEN Teil der Kommandozeile aus.

Beispiel:

```
SURE[Wollen Sie das File wirklich löschen?]:C:delete [f]
```

Dieses Kommando öffnet einen Requester mit der Frage "Wollen Sie das File wirklich löschen?" und den Auswahlmöglichkeiten "Ja" und "Nein".
Wählt man "Nein", dann geschieht nichts.
Wählt man "Ja", dann wird "C:delete [f]" ausgeführt.

Natürlich können Sie auch alle Argumente/Kommandos in einer Kommandozeile miteinander kombinieren.

Beispiel:

```
SURE[File wirklich umbenennen?]c:rename [f] REQF[Geben Sie mir einen neuen Namen]
```

1.17 Zukünftige Erweiterungen

Was ich gerne in der nächsten Programmversion noch einbauen würde:

- o den Begriff der 'Familie' von Klassen:
Z. B., daß die 'Bilder'-Familie GIF, IFF, TARGA, JPEG... enthält und das man sich dann durch einen 'Familien'-Filter im Verzeichnis-Requester Bilder, Sounds... usw. heraussuchen kann.
- o Unbegrenzte Anzahl von Pfad-Knöpfen.
- o Die Verwendung von WB-Fonts bei ClassAction.
- o Und natürlich alles das, wonach Sie mich fragen :)

1.18 Über

Den folgenden Personen möchte ich danken:

- o Mireille (für ihre Geduld...)
 - o Philippe Thomas (für Ideen, Hilfe, Beta-Testing, das Französische Guide und für die Verwendung von UUT für das Installations-Programm)

Hey Phil, wenn Du das liest, dann klick
~hier~
 - o Richier Pierre (für das MagicWB-Icon von ClassAction)
 - o Jean Michel and Georges (für Beta-Testing auf einem A4000/40)
 - o Obvious Implementations Corp (für Dice C Pro)
 - o Nico Francois für die ReqTools.library
 - o Georg Hörmann für the *GROßARTIGE* xfdmaster.library
 - o Allen Usern die mich auf Bugs und wegen Ihrer Ideen angesprochen haben
 - o Allen registrierten Usern
-

1.19 Der Autor

Sie können mich über die folgenden Adressen erreichen:

Gasmi Salim
6, rue des Hironnelles

67380 Lingolsheim
France

E-Mail: salim@panoramix.sdv.fr
salim@zeus.pandemonium.fr

1.20 Benutzung von ClassAction

Die Benutzung von ClassAction ist WIRKLICH einfach.

Wählen Sie nur ein File über den mitgelieferten Requester an. Sie können mit dem Knopf 'Mutterv.' oder über die Zeile <Parent> in dem Requester in das Mutterverzeichnis wechseln. Die rechte Maustaste zeigt Datenträger und Assigns.

Nach dem Auswählen sehen Sie in der rechten Aufzählung die File-Klasse und die dazugehörigen festgelegten Aktionen. Jetzt brauchen Sie nur noch die gewünschte Aktion zu wählen...

Wenn Sie ein File doppelt anklicken, wird die erste definierte Aktion durchgeführt.

Sie können das Fenster mit dem kleinen Zoom-Gadget in der rechten Fensterecke vergrößern/verkleinern.

Um das Programm zu beenden, klicken Sie auf den 'Ende'-Knopf.

Um das Fenster in ein AppIcon zu verwandeln, brauchen Sie nur das Fenster zu schließen. Wenn Sie das AppIcon doppelt anklicken, öffnet sich wieder das Fenster.

Wenn ClassAction 'appiconifiziert' ist können Sie Files daraufziehen, dann wird der File-Typ festgestellt.

Wenn die Klasse nur eine definierte Aktion hat, wird diese ausgeführt.

Ansonsten öffnet sich ein Fenster mit einer Liste aller möglichen Aktionen. Aus diesen können Sie dann eine auswählen.

Das ist alles, einfach nee???

1.21 Technische Informationen

ClassAction wurde 100% mit DICE C 3.0 programmiert.

Verschiedene Informationen:

Das prefs-File ist ein ASCII-File namens: ENVARC:ClassAction.prefs
Die Haupt-Aktionen werden in ENVARC:ClassAction_Gen.prefs gespeichert.

ClassAction läuft als ausführbares Programm unter ClassAction_RunTask
und wird in T: gespeichert.
Damit werden WB-Tasks durchgeführt.

Die Geschwindigkeit der Filerequester ist von den verwendeten Sortier-
algorithmen abhängig ('recursive tree sorting' namens 'Tri du Chat').

Libraries-Informationen:

Folgende ROM-Libraries werden benutzt:

exec.library	V37+
dos.library	V37+
intuition.library	V37+
graphics.library	V37+
gadtools.library	V39+
workbench.library	V37+
utility.library	V39+

Folgende DISK-Libraries werden benötigt:

rexxsyslib.library	V39+
commodities.library	V37+
asl.library	V39+
icon.library	V37+
retools.library	V38+

Folgende DISK-Libraries werden benutzt wenn sie vorhanden sind:

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

Wie ClassAction eine Klasse erkennt:

- 1- Testet, ob der Filename identisch mit dem MatchName einer definierten Klasse ist.
 - 2- Testet, ob es zu diesem File identische Offsets in einer definierten Klasse gibt.
 - 3- Entpackt das File mit der xfdmaster.library.
-

- 4- Testet, ob es in einer definierten Klasse identische Offsets zu dem im Puffer entpackten File gibt.
- 5- Testet, ob das File ein ASCII-File ist (wenn das ASCII-Offset-Kommando existiert).

Wenn alles fehlschlägt, wird das File als 'Unbekannte Klasse' aufgeführt.

1.22 Installation

Installation des Programmes:

Normalerweise sollten Sie das Installer-Skript verwenden, daß in diesem Archiv enthalten ist und dann den Anweisungen folgen.

Wenn Sie nicht über den Installer verfügen, sollten Sie folgendermaßen vorgehen:

- Kopieren Sie ClassAction, ClassActionPrefs und das Icon wohin Sie es möchten.
- Kopieren Sie das .prefs-Files nach ENVARC:
- Kopieren Sie das ClassAction.guide wohin Sie es möchten.

Das war's auch schon...

1.23 Registrierung

Wenn Sie diese Zeilen lesen, werden Sie sich darüber wundern, warum Sie ein registrierter ClassAction-User werden sollten.

Aber lassen Sie mich erklären, warum Sie sich registrieren lassen sollten:

Zu allererst einmal, um den besten Computer der je gebaut wurde zu unterstützen; weil die Zukunft des Amigas von der zukünftigen Software abhängt; und nicht zuletzt weil Sie, wenn Sie einen Programmierer unterstützen, auch Ihren Computer und seine Zukunft fördern!!!

Außerdem auch, weil ich meine ganze Freizeit darauf verwendet habe, Shareware-Programme zu entwickeln.

Wenn Sie sie benutzen, warum sollten Sie mir dann nicht auch die Registrierungs-Gebühr zukommen lassen?

Dies würde mich darin unterstützen, auch in Zukunft weiterhin Shareware-Programme zu entwickeln.

Diese Shareware-Version ist 100% benutzbar, nichts wurde gesperrt.

Aber wenn Ihnen das Programm wirklich gefällt, lassen Sie sich
REGISTRIEREN!!!

Die Registrierungs-Gebühr beträgt (entsprechend Ihrer Landes-
währung):

10 US\$ or 20 DM or 50 FF.

Senden Sie die Gebühr am meine
~Adresse~
und Sie werden
eine registrierte Version der aktuellsten Ausgabe von ClassAction
erhalten.

Vergessen Sie bitte nicht Ihre VOLLSTÄNDIGE Adresse anzugeben
(inclusive dem Land).

Vielen Dank bereits im voraus für Ihre Unterstützung!

Ihr

Salim

1.24 Rechtliches

Copyright

ClassAction und ClassActionPrefs sind Copyright © 1994-1995
bei Gasmi Salim.

ClassAction ist ein Shareware-Programm. Das Paket darf in keiner
Weise verändert oder zu kommerziellen Zwecken genutzt werde, ohne
das die ausdrückliche schriftliche Zustimmung des Autors vorliegt.
Die Copyright-Bestimmungen müssen gewahrt bleiben.

Garantie

Es wird keine Verantwortung oder Haftung für Schäden übernommen,
die aus oder durch die Benutzung dieses Programmes entstehen. Jede
Benutzung erfolgt auf Ihr eigenes Risiko. Das Programm wird "so wie
es ist" geliefert, ohne jede Garantie für die Funktionsfähigkeit
oder Genauigkeit des Programms und der Dokumentation. Die Dokumentation
sollte richtig und vollständig sein, aber der Autor behält sich das
Recht vor, die Software und/oder Dokumentation ohne vorherige
Ankündigung zu ändern.

1.25 Geschichte

ClassAction-Geschichte V 2.8 (c) Salim Gasmi

25/09/95 : V2.8

- ClassAction verfügt nun über eine VOLLSTÄNDIGE Commodity-Unterstützung.
- ToolType APPSTART kann auf HIDE (versteckt) gesetzt werden (APPSTART=HIDE), wenn Sie möchten, daß ClassAction versteckt startet.
- ToolType CX_HOTKEY hinzugefügt.
- Tooltype PUBSCREEN hinzugefügt, damit Public-Screens verwendet werden können.
- Der interne Fileselektor von ClassAction gab nicht den gesamten Speicher zurück, den er belegt hatte, jetzt behoben.

11/09/95 : V2.75

- Die REQD-, REQF-, REQV-Kommandos sind in Wirklichkeit nicht inkompatibel mit dem AppIcon-Modus, aber inkompatibel mit einigen Programmen wie MagicMenu (Pech für mich, daß ich MagicMenu benutze...).

Eine Menge User haben sich darüber beklagt, daß ClassAction automatisch zum REQT-Kommando wechselt, selbst wenn sie gar kein inkompatibles Programm verwenden.

Ich habe das ToolType (REQBUG) hinzugefügt, damit die User wählen können, ob automatisch in REQT gewechselt werden soll oder nicht.

03/09/95 : V2.7

- Kommandos [b],[x],[B],[X],[F],[S] hinzugefügt.
 - Die Aktion Arexx-Kommando hinzugefügt.
 - Einige Bugs in den Renderrountinen behoben.
 - Bug beim gleichzeitigen Ziehen mehrerer Icons auf das AppIcon behoben.
 - Die REQD-, REQF-, REQV-Kommandos werden nun in REQT geändert, wenn man sie vom AppIcon oder von Arexx aus benutzt, da sie nur mit dem Fenster-Modus kompatibel
-

sind.

28/08/95 : V2.6

- ClassAction-Fenster ist nun skalierbar.
- ClassAction und ClassActionPrefs verwenden jetzt die ReqTools.library.
- REQV-Kommando zum Auswählen der Datenträger hinzugefügt.
- REQT-Kommando zum Auswählen der Texte hinzugefügt.
- ToolTypes WINX und WINY hinzugefügt.
- Lernen-Requester hat jetzt einen 'Alle'-Knopf.
- Das ausgewählte File erscheint nun in allen REQ-Requestern.
- ClassAction verwendet jetzt ein Keyfile für die registrierte Version, das in S: abgelegt wird.

17/07/95 : V2.5

- ClassAction und ClassActionPrefs sind jetzt lokalisiert und ein französisches Guide ist im Paket enthalten.
- Lernen-Funktion zur ClassActionPrefs hinzugefügt.
- Um die erste Aktion für ein File anzuwählen muß man es nun doppelt anklicken, statt es nochmals anzuwählen.
- Haupt-Aktionen laufen nun synchron und durchsuchen das zugehörige Verzeichnis.
- ToolType AUTOSELECT wird nicht mehr verwendet.
- REQ-Requester öffnen sich nun im aktuellen Verzeichnis.
- Einige Verbesserungen durchgeführt.

12/06/95 : V2.1

- Die eingebaute 'Haupt-Aktionen'-Klasse hinzugefügt.
 - SURE[]-Kommando hinzugefügt.
 - ASCII[]-Offset-Kommando zum Erkennen von ASCII-Files hinzugefügt.
-

- Arexx-Kommandos AppIconify, Show, Status, GetClass hinzugefügt.
- 'Running-Task-System' geändert, ich verwende jetzt Systemtags(). Tmp.files werden nicht länger benötigt.
- Sie können nun die Anzeigzeit im CLI-Modus wählen.
- 'string' und "string" für die Offset-Definitionen hinzugefügt.
- ToolType OUTPUT ist jetzt veraltet und wird nicht mehr verwendet, als Ersatz wird jetzt das neue ToolType CLISIZE benutzt.
- Aktions-Requester sind nun richtig bemessen, erscheinen am Mauszeiger und verwenden den vordersten Publicscreen.
- Austausch der Knöpfe 'Benutzen' und 'Sichern' und hinzufügen von 'Abbruch' um dem 'Amiga-Prefs-Look' zu folgen. Den Knopf 'über' als '?' in die rechte obere Ecke versetzt.
- ClassActionPrefs Cyclegadgets waren nicht 100% systemfreundlich und einige Patches wie Cycle2Menu verursachten Fehler mit ClassActionPrefs; das ist jetzt behoben.
- Erkennungscode und die Inforoutine überarbeitet: sie sind bis zu 400% schneller.
- Farbfehler bei hervorgehobenen Zeilen in den Anzeigelisten behoben.
- ClassAction blockiert nicht länger den Workbenchschirm wenn es 'appiconifiziert' ist.
- Einen blöden Fehler in ClassActionPrefs gefunden und behoben.
- Die rechte Maustaste zeigt Assigns nur noch, wenn sich der Mauszeiger im Requester befindet und nochmaliges Drücken der rechten Maustaste bringt einen zurück zum Verzeichnis.
- ClassAction erinnert sich nun an die Fensterposition.

23/05/95 : V2.00 (Major Update)

- ClassAction hat jetzt ein AppIcon.
 - ClassAction ist jetzt ein Commodity.
 - ClassAction hat jetzt einen Arexx-Port.
 - ClassAction verwendet jetzt den Standart-WBFont.
-

- Exec-Modus 'Arexx' hinzugefügt.
- Unterschiedliche Farben für Verzeichnisse/Files.
- Hoch-/Runter-Knöpfe zu ClassActionPrefs hinzugefügt.
- 'Benutzen'-Knopf zu ClassActionPrefs hinzugefügt.
- Klassen werden jetzt sortiert im ClassActionPrefs-Anzeigefenster dargestellt.
- ToolTypes APPSTART, ICONNAME, ICONX, ICONY, CX_PRIORITY, WBFONT, OUTPUT, ICONFILE hinzugefügt.
- Wenn das Fenster iconifiziert ist, hat es jetzt die zum vorgegebenen Screen-Font passende Größe.
- Neues Speicherformat (CASF20).
- Suffix/Prefix-Knopf entfernt. Ersetzt durch das MatchName-feld, das alle Wildcards akzeptiert.
- Config-File nach ENVARC: verschoben.
- Ein Installer wird jetzt mit dem Paket mitgeliefert.
- Einige Code-Optimierungen vorgenommen.

05/05/95 : V1.43

- Requester-Code überarbeitet, ist jetzt 5% schneller.
- ClassAction verwendet jetzt WBStartup-Strukturen um nach seinem Namen zu suchen und kann umbenannt werden.

02/05/95 : V1.42

- Modus 'No Cli' zu ClassActionPrefs hinzugefügt.
- "" wird immer den Filenamen hinzugefügt auch wenn es nicht benötigt wird, das ist einfacher für AREXX-Skripts.

06/04/95 : V 1.4

- ToolType HEIGHT hinzugefügt.
 - Einige Code-Optimierungen durchgeführt.
-

22/02/95 : V 1.31

- Wenn man zweimal auf das gleiche File klickt, wird die erste Aktion durchgeführt (das ist schneller, als wenn man die erste Aktion per Hand auswählen muß).
- Diese Version ist jetzt ShareWare und man muß sich registrieren lassen um die registrierte Version zu erhalten.

15/01/95 : V 1.3

- Die interaktiven Kommandos REQD[] und REQF[] hinzugefügt.
- Copy-Knopf zu ClassActionPrefs hinzugefügt.
- Einige Verbesserungen gemacht.
- Betatester berichten, daß diese Version wirklich stabil läuft.

25/11/94 : V 1.22

- Erste Veröffentlichung.
- 'Info'-Knopf hinzugefügt.
- ClassAction sperrte das Verzeichnis in dem es las [Locking()] ohne es wieder freizugeben [UnLocking()]...
KORRIGIERT.
- Code-Optimierung.
- Einige kleine Fehler behoben.

08/11/94 : V 1.21

- Programm stürzte ab, wenn keine Diskette eingelegt war...
KORRIGIERT.
- Verzeichnis/Name Fehlerkorrektur.
- <.>-Teil entfernt, wenn es sich um ein Wurzelverzeichnis handelt.

07/11/94 : V 1.2

- Neue Schnittstelle, ich verwende jetzt meine eigenen schnelleren File-Requester.
-

- ToolTypes STARTDIR & DRIVE1 bis DRIVE11 hinzugefügt.

01/11/94 : V 1.1

- Ab jetzt wird die xfdmaster.library zum Erkennen und Entpacken der Files verwendet.
- Konfiguration durch ToolTypes hinzugefügt (DECRUNCH,AUTOSELECT).
- 'Unbekannte Kasse' ist jetzt fest eingebaut und hat eine unbegrenzte Anzahl von Aktionen.
- Neues Speicherformat (CASF11).

16/10/94 : V 1.0

- Neues Speicherformat (CASF10).
- Fenster hat nun einen Zoom-Knopf.
- Eine ganze Menge Klassen-Definitionen hinzugefügt.

10/10/94 : Beta-Version

- Fehler in den tmp-Files behoben.
- Fehler in der Offset-Steigerung behoben.
- Verwende jetzt die asl.Library für die File-Requester.

01/10/94 : Alpha-Version

1.26 Die Arexx-Kommandos

ClassAction hat einen Arexx-Port namens: ClassAction.01

Dies ist die Liste aller Arexx-Kommandos:

=====

Quit

Beendet ClassAction...

Use

Veranlaßt ClassAction zum Laden des Prefs-File.

Ver

Gibt die Version von ClassAction aus.

Status

Gibt den gegenwärtigen Status von ClassAction aus.

Ausgabe 0 : ClassAction ist 'AppIconifiziert'.

Ausgabe 1 : ClassAction ist im Fenster-Modus.

AppIconify

Veranlaßt ClassAction sich zu 'AppIconifizieren'.

Wenn ClassAction bereits ein AppIcon ist, geschieht nichts.

Show

Veranlaßt ClassAction das Hauptfenster zu öffnen.

Wenn ClassAction bereits ein Fenster ist, geschieht nichts.

Load <FileName>

ClassAction versucht das File <FileName> zu laden und öffnet den Aktions-Requester, damit der User eine davon auswählen kann.

Wenn das File nicht existiert, geschieht nichts.

GetClass <FileName>

ClassAction versucht das File <FileName> zu laden und gibt die Kasse des geladenen Files aus

Wenn das File nicht existiert, geschieht nichts.

Aktion <FileName> <Aktions-Muster>

ClassAction führt die erste zugehörige Aktion für das File <FileName> aus.

Beispiel: Aktion ram:toto.lha extr

Das startet die erste Aktion mit dem String 'extr' im Namen mit dem File ram:toto.lha.

Dieses Muster ist nützlich, sie brauchen nicht mehr den genauen Namen der Aktion anzugeben, die Angabe eines Teils des Namens reicht bereits.

1.27 Was sollten Sie nach der Installation machen

Nach der Installation verfügen Sie über ein startbereites ClassAction mit einer ganzen Menge definierter Klassen, aber mit sehr wenigen dazugehörigen Aktionen (die meisten der festgelegten Klassen haben keine zugehörigen Aktionen).

Es liegt nun an Ihnen, Aktionen zu definieren, wie sie von Ihrer Systemkonfiguration und von Ihren bevorzugten Programmen benötigt werden.

Dann laden Sie nur noch ClassActionPrefs und richten es nach Ihren Vorstellungen und Wünschen ein...

Wenn Sie nicht wissen, wie Sie Aktionen und Klassen einrichten, dann sollten Sie diesen Guide lesen :)).

Nachdem Sie die Aktionen und Klassen festgelegt haben, brauchen Sie nur noch ClassActions Verhalten über seine ToolTypes festzulegen.

Es mag eine lange Zeit dauern, bis Sie eine 'tolle' Konfiguration erreicht haben.

Aber, wenn einmal alles geschafft ist, dann ist es GROßARTIG!!!

Und nun lesen Sie das Guide und viel Erfolg.

1.28 Message for 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 merci pour tous les appels téléphoniques ; parfois plus d'une heure à discuter des Hooks, SystemTagList, de bugs etc... et ce, même en période 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 Bekannte Fehler

Eine ganze Menge User haben mich auf eine skalierbare GUI angesprochen. Und ich habe mein Bestes getan, um diese V2.6+ Version mit einem skalierbaren Fenster auszustatten, aber es ist ein kleines bisschen kompliziert mit GadTools.

Ganz selten, wenn Sie die Größe Ihres Fensters verändern, kann es passieren, daß sich der Refresh 'aufhängt' und Sie müssen dann das Fenster schließen, um diese 'Aufhängung' wieder zu beseitigen.

Ich weiß wirklich nicht warum, da ich das Problem bei mir selbst noch nie festgestellt habe. Die Betatester haben mir von diesem Problem berichtet, das sie nur 2 mal in einem einmonatigen Test festgestellt haben.

Trotzdem werde ich mein Bestes geben, um diesen Bug zu finden und zu beheben. Aber ich werde nicht einer Menge Usern eine skalierbare GUI vorenthalten wegen eines Bugs, der 2 mal in einem Monat auftaucht.

Einige Programme wie MagicMenu führen zum Absturz, wenn REQ-Kommandos über das AppIcon verwendet werden. Sie können das ToolType REQBUG verwenden, um die REQ-Befehle abzuschalten.