

# **ClassAction**

Martin R. Gasmi

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> ClassAction		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Martin R. Gasmi	July 31, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ClassAction</b>	<b>1</b>
1.1	ClassAction 4.2 Guide . . . . .	1
1.2	Index . . . . .	2
1.3	Was ist ClassAction ? . . . . .	3
1.4	Mehr über ClassAction . . . . .	4
1.5	Systemanforderungen . . . . .	4
1.6	Konfiguration . . . . .	5
1.7	Optionen . . . . .	6
1.8	Befehle . . . . .	10
1.9	Klassen . . . . .	12
1.10	Benutzung der Lernfunktion . . . . .	12
1.11	Was ist eine Klasse . . . . .	13
1.12	Eine neue Klasse erstellen . . . . .	14
1.13	Eine neue Aktion definieren . . . . .	17
1.14	Startmodi . . . . .	18
1.15	Variablen . . . . .	20
1.16	Kommandos . . . . .	21
1.17	Zukünftige Verbesserungen . . . . .	22
1.18	Über . . . . .	22
1.19	Der Autor . . . . .	24
1.20	Benutzung von ClassAction . . . . .	24
1.21	Die Knöpfe . . . . .	26
1.22	Das Menü . . . . .	27
1.23	Technische Infos . . . . .	27
1.24	Installation . . . . .	28
1.25	Rechtliches . . . . .	29
1.26	Geschichte . . . . .	29
1.27	Die Arexx Kommandos . . . . .	34
1.28	Die Archiv-Unterstützung . . . . .	36
1.29	FTPMount . . . . .	37
1.30	Was sollte man nach der Installation tun . . . . .	39
1.31	Mitteilung für Philippe THOMAS . . . . .	39
1.32	Bekannte Fehler . . . . .	40

# Chapter 1

## ClassAction

### 1.1 ClassAction 4.2 Guide

ClassAction  
Version 4.2

Für Anfänger

- o ~Was~ist~ClassAction~~~~~ Dies zuerst lesen
- o ~Systemanforderungen~~~~~ Was du brauchst
- o ~Rechtliches~~~~~ Was du darfst

Installation und Konfiguration

- o ~Installation~~~~~ Wie installiert man ClassAction
- o ~Nach~der~Installation~~~~~ Klassen und Aktionen definieren
- o ~ClassActionPrefs~~ ~~~~~ ClassAction konfigurieren
- o ~Optionen ~~~~~ Parameter ändern
- o ~Befehle ~~~~~ Die Oberfläche anpassen
- o ~Klassen ~~~~~ Klassen und Aktionen konfigurieren

ClassAction benutzen

- o ~Allgemeine Hinweise ~~~ Wie benutzt man dieses Programm
- o ~Die Knöpfe ~~~ Was sie bedeuten und tun
- o ~Das Menü ~~~ Alle Menüpunkte

Zusätzliche Funktionen

- o ~Arexx~Kommandos~~~~~ Der ARexx Port und Kommandos
- o Archiv~Unterstützung ~~~ Was man mit \*.lha und \*.zip-Dateien machen kann
- o FTPMount ~~~ Wie man mit ClassAction FTP nutzt

Über dieses Programm

- o ~Geschichte~~ Seit dem Anfang...
- o ~Die~Autoren~~~~~ Wer dieses Programm schrieb
- o ~Grüße~~~~~ An alle, die geholfen haben!
- o ~Bekannte~Fehler~~~~~ Niemand ist perfekt
- o ~Zukünftige~Verbesserungen~~ Was in der Zukunft noch kommen könnte

Index

## 1.2 Index

### A

Aktionen  
Anforderungen  
ARexx Befehle  
Archive  
Autoren

### B

Befehle  
Befehle (intern)

### C

ClassAction  
ClassActionPrefs

### D

Devices  
Directory Opus

### E

Einstellungen  
Einstellungen (Menüpunkt)

### F

Fehler  
FTP  
FTPMount

### G

Garantie  
Geschichte  
Grüße

### I

Installation

### K

Klassen  
Knöpfe  
Konfiguration

---

L

Lha Dateien  
Lizenz

M

Menü

O

Optionen

S

Start-Modi  
Systemanforderungen

Z

Zip Dateien  
Zukünftige Verbesserungen

## 1.3 Was ist ClassAction ?

Was ist ClassAction ?

ClassAction ist ein kleines Programm, das das Leben aller Festplattennutzer vereinfacht.

Wenn du eine Festplatte besitzt, befinden sich dort auch eine Menge Dateien: ausführbare, Module, Bilder, Quellcodes, Animationen, Sounddateien...

ClassAction stellt für dich die Art der gewählten Datei fest und gibt eine Liste der möglichen Aktionen mit dieser Datei an.

Wenn du zum Beispiel ein GIF Bild auswählst, wird es als solches erkannt, und eine Liste der möglichen Aktionen, wie 'Anzeigen' oder 'Ändern' erscheint.

ClassAction ist auch ein vollständiger Dateimanager mit vielen Optionen. Kopieren, bewegen, umbenennen, löschen, Icons hinzufügen, Informationen anzeigen, ...  
Probier es einfach aus, und du wirst alle Möglichkeiten kennenlernen!

ClassAction ist sehr gut konfigurierbar, man kann eigene Klassen und Aktionen hinzufügen.  
Aktionen benutzen externe Programme, kann man also seinen bevorzugten GIF-Anzeiger einbinden (oder ähnliches).  
Außerdem kann man das Verzeichnismenü selber auswählen und eigene Icons und Tastaturbefehle definieren.

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

---

ClassAction nutzt die `xfdmaster.library`, um Dateien zu entpacken. Durch diese Eigenschaft können sogar gepackte Klassen erkannt werden.

ClassAction nutzt sehr wenig Speicher und CPU Zeit.

Wenn du dir bei der Konfiguration von ClassAction ein wenig Zeit nimmst, kannst du alles damit tun !!!  
Es ist ein einfacher Weg, mit Dateien umzugehen.

Bevor du es löschst, probier es aus !!!!!!!

Siehe auch

~Mehr~über ClassAction .

## 1.4 Mehr über ClassAction

Mehr über ClassAction

ClassAction und ClassActionPrefs sind  
© 2001 Martin R. Elsner und Salim Gasmi.

Dieses Paket ist nun Mailware, also schick dem Autor bitte eine Email, wenn du das Programm regelmäßig benutzt.

PD-Anbieter dürfen das ClassAction Paket in ihre Sammlung aufnehmen, wenn sie mich davon unterrichten.

Ich hoffe, du findest es nützlich.  
(ich schon :) )

Siehe auch

Technische~Infos~.

## 1.5 Systemanforderungen

Anforderungen

- o Amiga OS 3.5 oder höher
- o eine Festplatte  
(ClassAction ist ohne nutzlos)

Empfohlen:

---

- o Die Programme "Lha" und "Zip"
- o Eine hohe Auflösung wie 1024 x 768
- o Höhere Geschwindigkeit, z.B. ein 68060 ...

## 1.6 Konfiguration

### Konfiguration

Die gesamte Konfiguration wird mit ClassActionPrefs bearbeitet. Du musst weder Tooltypes noch Konfigurationsdateien bearbeiten, starte einfach ClassActionPrefs - z.B. aus dem ClassAction-Menü - und ändere die Einstellungen.

Du kannst zwischen drei Abschnitten wählen:

- o Optionen

Einige globale Eigenschaften des ClassAction-Fensters, der Art, wie ClassAction funktioniert und aussieht.

- o Befehle

Die Icons oben und unten im Fenster und die Tastatur-Befehle.

- o Klassen

Die Klassen und Aktionen - der wichtigste Teil.

Das Menü besitzt Punkte, wie man sie von Standard-Voreinstellungsprogrammen kennt:

### Projekt:

- o Öffnen... - irgendeine Einstellungsdatei öffnen
- o Speichern als... - die Einstellungsdatei irgendwo speichern

Da ClassAction mehrere Dateien benötigt, beziehen sich diese Befehle nur auf die aktive Seite (Optionen/Befehle/Klassen).

### Vorgaben:

- o auf Vorgaben zurücksetzen - noch nicht implementiert
- o auf zuletzt gespeichertes - lädt die Einstellungen, die zuletzt in ClassActionPrefs gespeichert wurden
- o auf vorherigen Stand - setzt die Einstellungen auf den Stand beim Start von ClassActionPrefs zurück

Wenn du alle Änderungen durchgeführt hast, wähle Speichern, um die Einstellungen zum Standard zu machen, oder Benutzen, um sie nur bis zum nächsten Neustart zu testen. Abbrechen oder das Schließsymbol werden alle Änderungen



verwerfen.

## 1.7 Optionen

Optionen

Manche Funktionen in ClassAction sind konfigurierbar. Um sie zu ändern, starte ClassActionPrefs, änder die Optionen und wähle "Speichern" oder "Benutzen".

Anmerkung: Wenn du Optionen geändert hast, während ClassAction läuft, können manche Optionen nicht direkt angepasst werden; sie sind hier mit einem \* gekennzeichnet. ClassAction wird alle Änderungen durchführen, wenn es das nächste Mal gestartet wird.

Manche Optionen können auch in ClassAction geändert werden. Man findet sie im Einstellungsmenü .

Hier eine Liste aller Optionen. Manche sind Schalter und werden abgehakt, wenn man auf sie klickt, manche haben Werte, die mit einem Doppelklick geändert werden können.

Allgemein:

Starte als Icon \*

Wenn du diese Option anschaltest, startet ClassAction als AppIcon. Das ist z.B. nützlich, um ClassAction direkt in WBStartup zu starten.

Verbergen durch Schließgadget

MUI-Anwendungen können durch einen Klick auf das Schließsymbol ikonifiziert werden. Da ClassAction einst eine solche war und deine Gewohnheiten nicht stören möchte ;) kannst du dieses Verhalten aktivieren.

Ersetzung bestätigen

Wenn man auf 'Kopieren' oder einen ähnlichen Knopf klickt, würde möglicherweise eine alte Datei mit dem gleichen Namen überschrieben. Wenn diese Option aktiv ist, wirst du dann gefragt, ob die alte Datei wirklich gelöscht werden soll.

Platz vor dem Kopieren prüfen

Schalte dies an, um zu überprüfen, ob die ausgewählten Dateien auf das Ziellaufwerk passen, bevor ClassAction mit dem Kopieren beginnt.

Hilfe zeigen

Wenn du nicht genau weißt, welches Icon welchem Zweck dient,

dann aktiviere diese Option, und ein kleiner Hinweis wird am unteren Rand angezeigt.

#### Öffentlicher Bildschirm

Vielleicht benutzt du einen anderen Schirm und willst ClassAction auf diesem starten. Wenn du seinen Namen kennst, gib ihn einfach hier an, und ClassAction und ClassActionPrefs werden diesen Bildschirm statt der Workbench benutzen. Wenn kein Name oder "Workbench" angegeben wird, wird wie gewohnt die Workbench genutzt.

#### Cli-Größe

Das ist die Größe des CLI Ausgabefensters, wenn ein Befehl über Cli ausgeführt wird.

Die Syntax ist : GERÄT:ObenX/ObenY/Breite/Höhe/Titel

Dies kann auch zu einem anderen gerät als CON: gesetzt werden oder eine andere Fensterposition und Dimension festsetzen.

#### WARNUNG :

Wenn dies auf einen falschen Wert gesetzt wird, öffnet ClassAction kein CLI.  
Ändere NICHTS, wenn du nicht genau weißt, was du tust.

Benutze \*\* NIEMALS \*\* Kommandos, wie AUTO, CLOSE, WAIT, da dies schon ClassAction tut.

Benutze \*\* NIEMALS \*\* Leerzeichen im Titel.

Voreingestellt ist :  
CON:0/0/640/100/ClassAction\_Output\_Window

#### Fenster:

##### Linker Rand, Oberer Rand

Damit kannst du den Abstand des ClassAction-Fensters vom linken und oberen Rand des Bildschirms angeben.  
Wenn beide auf -1 gesetzt werden, wird das Fenster zentriert.

##### Höhe, Breite

Gib hier die Höhe und Breite des Fensters an.  
Falls hier nichts steht, wird die Größe nach dem Fensterinhalt berechnet.

##### AppIcon linker Rand, AppIcon oberer Rand

Normalerweise erhält das AppIcon auf der Workbench die gleiche Position wie das Programm-Icon. Wenn dir dies nicht gefällt, gib diese Werte an. Ein Wert von -1 wird die Position nicht

verändern.

#### Popup Wartezeit

In manchen Fällen kannst du eine Aktion aus einem Popup-Menü wählen, das sich gewöhnlich nach einigen Sekunden schließt. Hier kannst du diese Verzögerung einstellen. Gib die Zahl der Sekunden an, die das Menü offen bleiben soll, oder einen Wert von -1, wenn du gar nicht willst, dass das Menü sich schließt (du kannst immer noch irgendwo hinklicken oder die rechte Maustaste oder irgendeine Taste drücken).

#### Knöpfe:

##### Mutterverzeichnis \*

Zeige und benutze den "Parent"-Knopf links vom Pfad des Quell- und Ziellisters, mit dem man ins übergeordnete Verzeichnis wechseln kann.

##### Pfad-Menü \*

Zeige und benutze das Pfadmenü rechts von Pfad in Quell- und Ziellister. Die Pfade kannst du in der Gruppe "Pfad-Menü" angeben.

##### Geschichte \*

Zeige und benutze den Geschichtsknopf rechts von Pfad in Quell- und Ziellister.

#### Lister:

##### Symbole anzeigen

Bilder sind ja schön, aber wenn du zu wenig Platz hast oder die Icons häßlich findest :( stell den Schalter aus. Dann solltest du "Verzeichnisse hervorheben" anschalten.

##### Benutze kleine Symbole

Wähle diese Option, um kleinere Bilder links von Dateien/ Verzeichnissen darzustellen.

##### Devices anzeigen

In den meisten Fällen werden dir Laufwerke und Assigns reichen - aber wenn du auch die Devices sehen möchtest, dann wähle diese Option.  
(Dies kann interessant sein, wenn du einen env:-Handler u.a. installiert hast.)

##### Verzeichnisse hervorheben

Schalte dies an, um Verzeichnisse in weißer Schrift

anzuzeigen. Sonst werden sie genauso schwarz wie die anderen Einträge dargestellt.

Zeige Verzeichnisse zuerst

Benutze diese Option, um die Sortierung der Dateilisten anzupassen. Angeschaltet heißt: zeige erst alle Verzeichnisse, dann die Dateien.

Datum als Text zeigen

Wenn dies aktiviert ist, werden manche Datumsangaben durch Text ersetzt, z.B. "Heute", "Montag" etc.

Auch Zeit anzeigen

ClassAction zeigt das Datum der letzten Änderung für jede Datei an, und zusätzlich die Uhrzeit, wenn diese Option aktiv ist.

Startpfad Quelllister

Dies ist das Verzeichnis, das im Quell-Lister angezeigt wird, sobald ClassAction startet.

Startpfad Ziellister

Das gleiche wie "Startpfad Quelllister" für den Ziel-Lister.

Icons:

Icons anzeigen

Wenn diese Option gesetzt ist, werden .info Dateien in Quell- und Ziellister angezeigt.

Icons erzeugen

'Verzeichnis erzeugen' erzeugt immer ein neues Verzeichnis, aber erzeugt nur ein Icon, wenn diese Option aktiv ist.

Icons kopieren

Wenn dieser Schalter aktiv ist, werden 'Kopieren' und ähnliche Knöpfe immer auch die Icons der gewählten Dateien kopieren. Dies ist vor allem nützlich, wenn "Icons anzeigen" nicht aktiv ist.

Pfade:

Pfad zum AppIcon

Dieser Pfad führt zum AppIcon von ClassAction.  
D.h., daß man sein eigenes AppIcon wählen kann:  
Wenn man diesen Pfad auf eine Piktogrammdatei setzt, benutzt

---

ClassAction dieses Icon als AppIcon.

Du brauchst die .info Endung nicht anzugeben.

Beispiel : Sys:icons/head  
nutzt das Piktogramm head.info aus dem Verzeichnis  
Sys:Icons/ als AppIcon.

Wenn das Laden des Piktogramms fehlschlägt, wird das voreingestellte benutzt (ClassAction.info).

Pfad zur Anleitung

Wenn du diese Anleitung in einem anderen Verzeichnis als HELP: speicherst, dann musst diesen Ort angeben.  
Beispiel: sys:docs/classaction.guide

Pfad zu ClassActionPrefs

ClassActionPrefs speichert auch einen weiteren Pfad in den Einstellungen: seinen eigenen Pfad!  
Weil es weiß, wo es sich befindet, musst du es nicht ändern ;)

Pfad-Menü:

Du hast 20 freie Plätze für deine eigenen Pfade, die im Pfad-Popupmenü angezeigt werden.

Klicke doppelt auf einen dieser Einträge. Zuerst wirst du nach einem Verzeichnis gefragt, das aufgelistet werden soll, dann nach einem Namen, der als Titel im Popupmenü angezeigt wird.

Du kannst auch Einträge frei lassen, die dann ausgelassen werden.

Siehe auch

Einstellungsmenü

## 1.8 Befehle

Befehle

Du kannst die Icons oben und unten im Fenster sowie Tastaturbefehle ohne Icon konfigurieren.

Starte einfach ClassActionPrefs und schau dir die vordefinierten Befehle an, um einen Überblick zu bekommen.

Mit "Hinzufügen" kannst du einen neuen Befehl erzeugen.  
Dann kannst du ihn mit den Pfeilen an die richtige Stelle bewegen.

---

## Name und Hilfe

Gib einen kurzen Namen an, der nur in ClassActionPrefs gilt und für ClassAction keine Bedeutung hat. Diesen Text wirst du in der linken Liste sehen.

Gib einen Hilfstext an, der in ClassAction angezeigt wird, wenn "Hilfe zeigen" angeschaltet ist.

## Befehl

Der wichtigste Teil ist der Befehl selbst:

entscheide dich zunächst, ob du etwas über Cli (CLI), über Cli ohne Fenster (No CLI), über die Workbench (WB), als ARexx-Skript (ARexx) oder als interner Befehl (Internal) starten möchtest, siehe auch Startmodi .

Wenn du die Iconleisten konfigurierst, wirst du vielleicht zwischen zwei Icons einen Freiraum lassen wollen; erzeuge dazu einfach ein Icon mit dem Typ SPACE.

Wähle jetzt den Befehl aus. Du hast mehrere Möglichkeiten:

- o Gib den Befehl direkt im Eingabefeld ein.
- o Wähle "Laden" und suche dann das Programm, das du einfügen möchtest.
- o Wenn du einen internen Befehl benutzen möchtest, wähle aus dem Menü "Interne Befehle" denjenigen Befehl aus.

Zusätzlich kannst du einige interne Variablen benutzen, die durch die ausgewählte Datei, die angezeigten Verzeichnisse, eine Zeichenkette oder eine Datei, die vom Benutzer ausgewählt wird, ersetzt werden ... Siehe Variablen für eine vollständige Beschreibung.

## Icon

Wenn du den Befehl als Icon im ClassAction-Fenster sehen möchtest, musst du eine Icon-Datei angeben. Gib dazu den Namen einer Datei an, die ein Icon besitzt (ClassAction wird die Endung ".info" löschen, wenn du sie angibst).

Du kannst dir das Icon anschauen, indem du "Vorschau" wählst, oder sogar die gesamte Iconleiste, indem du zuerst die Gruppe und dann "Vorschau" wählst.

## Taste

Schließlich kannst du noch eine Taste für diesen Befehl auswählen. Falls es ein Tastaturbefehl ist, ist dies die einzige Möglichkeit, den Befehl aufzurufen.

Du hast zwei Möglichkeiten: ASCII oder RAWKEY.

ASCII akzeptiert einen anzeigbaren Buchstaben wie 'a', 'W' or ':' und unterscheidet zwischen Groß- und Kleinbuchstaben.

RAWKEY identifiziert die Nummer der gedrückten Taste. Damit kannst du Tasten wie F1,...,F10 oder ESC,DEL,TAB,RETURN ... definieren.

Wenn du die Nummer der Taste nicht kennst ;) drück auf "Wähle RAW

---

Taste" und dann die Taste, die du auswählen möchtest. Die Nummer wird dann angezeigt.

Sei vorsichtig, denn du kannst andere Befehle damit überschreiben; die Hilfe-Taste und die lateinischen Buchstaben werden schon von ClassAction benutzt, aber können durch deine eigenen Befehle überschrieben werden.

Siehe auch

Interne Befehle  
Startmodi  
Variablen

## 1.9 Klassen

Klassen

Neue Klassen und Aktionen zu definieren ist der Hauptzweck des Programms. Dazu mußt du das Programm ClassActionPrefs nutzen.

Die Benutzung sollte einfach sein... also los !!

Das Fenster ist in zwei Teile geteilt : Klassen & Aktionen.

Wenn eine Klasse gewählt ist, werden die zugehörigen Aktionen im Aktionsteil dargestellt.

Um eine Klasse hinzuzufügen oder zu löschen, einfach die entsprechenden Knöpfe drücken.

Genauso geht's bei den Aktionen...

~1.~Was~ist~eine~Klasse~~~~~

~2.~Eine~neue~Klasse~erstellen~~

~3.~Eine~neue~Aktion~definieren~

~4.~Die~Lernfunktion~~~~~

## 1.10 Benutzung der Lernfunktion

Benutzung der Lernfunktion

Die Lernfunktion soll beim Erstellen neuer Klassen helfen.

---

Wenn man eine neue Klasse definiert, muß man deren Offset setzen, und es ist langwierig und langweilig, diesen herauszufinden.

Darum wurde diese wunderbare Funktion geschaffen !

Um sie zu nutzen, tue folgendes :

- 1 : Definiere deine Klasse normal, setze Namen und Klassennamen.
- 2 : Klicke den Knopf 'Lernen'.
- 3 : Wähle in dem folgenden Auswahlfenster soviele Dateien, wie du möchtest und sie der neuen Klasse angehören. Je mehr du wählst, desto besser wird das Resultat. ClassAction versucht nun, die Offsetdefinition zu finden. (um mehrere Dateien auszuwählen, nutze die Shift-Taste.)
- 4 : Nach der Analyse bekommst du ein Fenster mit den gefundenen Offsets.
- 5 : Ändere dies, wenn nötig, von Hand.
- 6 : Klicke auf 'Akzeptieren', um diese Offsets für deine Klasse zu nutzen, oder 'Abbruch', um abubrechen.

WARNUNG :

Wenn du die Lernfunktion nutzt, mußt du sicher sein, daß :

- \* ALLE GEWÄHLTEN DATEIEN DERSELBEN KLASSE ANGEHÖREN.
- \* ALLE GEWÄHLTEN DATEIEN NICHT GEPACKT SIND.

Das ist alles !!! Einfach, nicht ? .

## 1.11 Was ist eine Klasse

Was ist eine Klasse

Eine Klasse ist eine Familie von Dateien. Zum Beispiel können C Dateien als eine Klasse bezeichnet werden, als C Klasse.

Mit ClassActionPrefs kann man soviele Klassen definieren, wie man will, solange man diese unterscheiden kann.

Es gibt zwei Methoden, eine Klasse zu erkennen : Den Erkennungsnamen und den Inhalt der Datei.

Mittels des Erkennungsnamens kann man eine Datei anhand des Namens erkennen.

Offsets werden benutzt, um eine Datei anhand ihres Inhalts zu erkennen.

---



Es gibt drei eingebaute, nicht entfernbare, Klassen, die weiß in der Klassenliste auftauchen und nicht umbenannt werden können.

Die erste heißt "Unbekannte Klasse". Diese Klasse enthält alle Dateien, die ClassAction nicht erkennen kann.

Die zweite heißt "Allgemeine Aktionen". Diese Klasse enthält Aktionen, die in ALLEN anderen Klassen sind...

Interessant : falls du eine Aktion 'Kopieren' in allen Klassen hast, so könntest du diese in allen Klassen definieren, was aber lange dauert und anstrengt.

Besser ist es, diese in der Klasse "Allgemeine Aktionen" zu definieren, und sie gilt für alle Klassen.

Die allgemeinen Aktionen werden in der Aktionsliste weiß dargestellt und sind nur im ClassAction Fenster sichtbar, nicht im AppIcon oder im ARexx-Modus.

Die dritte heißt "Verzeichnis" und enthält die Aktionen für ganze Verzeichnisse.

## 1.12 Eine neue Klasse erstellen

Nur auf den Knopf 'Hinzufügen' im Klassenteil drücken und schon wird eine neue Klasse erstellt.

Eine Klasse hat 3 Merkmale :

- einen Namen
  - einen Erkennungsnamen
  - Offsets
- o Der Name ist einfach der Klassenname, wie du ihn wählst.  
WARNUNG : ein Klassenname darf nur einmal vergeben werden.
  - o Der Erkennungsname ist jede reguläre AmigaDos-Endung, wie :

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

(lies im AmigaDOS-Handbuch zum Thema Wildcards)

WARNUNG : Benutze nicht \* sondern #? als Wildcard.

Der Erkennungsname ist nicht einmalig, z.B. könnte toto.c in mehreren Klassen auftauchen.

Wenn du eine Klasse anhand des Erkennungsnamens definierst, mußt du dieses gut überlegen.

Beispiel : wenn du die GIF Klasse mit dem Erkennungsnamen #?.gif

---

definierst, werden alle Dateien mit der Endung .gif als GIF Dateien interpretiert.

Aber bist du sicher, daß alle deine GIF Dateien diese Endung haben und das ALLE Dateien mit der Endung .gif auch GIF Dateien sind ?

Darum solltest du den Erkennungsnamen nur in zwei Fällen nutzen :

- der Erkennungsname ist unabhängig von Groß- und Kleinschreibung (z.B.: #?.info ist ein guter Erkennungsname für die Klasse Piktogramm)
- wenn du keine Wahl hast (z.B.: wie kann man einen C-Quellcode erkennen, wenn nicht mit #?.c)

- o Andernfalls solltest du Offsets nutzen.

Ein Offset ist eine Stelle in einer Datei, wo man etwas Erkennbares finden kann.

Zum Beispiel beginnen GIF Bilder immer mit 'GIF' am Offset 0.

Es gibt drei Syntax, um Offsets zu definieren :

=====  
Syntax #1 :   Offset,HexString

Offset ist eine DEZIMALE Zahl, die den Offset enthält.  
HexString ist ein HEX-Sstring, der bei diesem Offset steht.

Beispiel 1 : 0,4f4a   heißt, daß die Datei mit den Bytes \$4f und \$4a an Position 0 beginnen muß.

Beispiel 2 : 9,448b3c heißt, daß bei Byte #9 \$44 \$8b \$3c gefunden werden sollten.

=====  
Syntax #2 :   Offset,'String'

Offset ist eine DEZIMALE Zahl, die den Offset enthält  
String ist eine ASCII-Zeichenkette, die bei dem Offset steht.

Beispiel 1 : 0,'GIF'   heißt, daß die datei mit 'GIF' beginnen muß.

Beispiel 2 : 9,'FuBar' heißt, daß man bei Byte #9 die Zeichenkette 'FuBar' finden muß.

---

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

Offset ist eine DEZIMALE Zahl, welche den Offset enthält.  
String ist eine ASCII-Zeichenkette, die bei diesem Offset stehen muß.

Beachte den Unterschied zur vorherigen Syntax, hier nutzen wir "  
statt '.

Es ist dasselbe Konzept, wie bei Syntax #2, aber der Vergleich  
ist UNABHÄNGIG VON GROß- ODER KLEINSCHREIBUNG.

Zum Beispiel beginnt eine AmigaGuide datei immer mit @database  
mit großen und kleinen Buchstaben.

Wenn du Methode #2 benutzt mit 0, '@database' wird ClassAction  
eine Datei mit @DATABASE nicht als AmigaGuide datei erkennen.

Es funktioniert aber mit Syntax #3 : 0, "@database"

=====

Man kann bis zu 5 Offsets in einer Klasse definieren. Eine Datei wird  
als der Klasse zugehörig erkannt, wenn alle Offsets übereinstimmen.  
Wenn die Datei in mehr als eine Klasse passt, wird die Klasse mit  
den meisten Offsets gewählt.

Bsp. : wenn Klasse X so definiert ist :

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

Werden alle Dateien, di mit 4a8b6c beginnen UND bei Byte #58 \$14  
haben als X deklariert.

Um mehrere Offsets zu deklarieren, nur den Knopf 'Offset #' drücken.

Anmerkung : Die ASCII Klasse

Es gibt ein eingebautes Offset Kommando : ASCII[]

Wenn du dieses Kommando in Offset #1 (d.h. Offset#1=ASCII[])  
einschreibst, werden ASCII Dateien ausgewertet.  
ClassAction probiert das, nachdem alles andere fehlschlägt. Darum  
werden Amigaguide Dateien im ASCII-Format trotzdem als AmigaGuide  
Dateien erkannt, wenn die AmigaGuide-Klasse definiert wurde.

Normalerweise sollte dich das nicht kümmern, da ich in der  
Standardeinstellungsdatei dieses schon vordefiniert habe.

---

## 1.13 Eine neue Aktion definieren

Wenn eine Klasse definiert ist, solltest du Aktionen dafür bereitstellen. Jede Klasse kann so viele Aktionen haben, wie du möchtest.

Klicke einfach auf den Knopf 'Hinzufügen' im Aktionsteil, um eine neue Aktion zu schaffen.

Eine Aktion hat 6 Merkmale:

- einen Namen
  - einen Laufmodus
  - eine Stapelspeichergröße (StackSize) (nur im CLI-Modus)
  - eine Verzögerung (nur im CLI-Modus)
  - ein Exec Kommando
  - ein RescanDir Zeichen (Verzeichnis wiedereinlesen)
- 
- o Name ist der Name der Aktion.
  - o Modus kann sein : ~CLI~, ~WB~, ~Kein~CLI~ oder ~ARexx~.
  - o Befehl ist eine gültige AmigaDOS Kommandozeile und kann verschiedene Parameter enthalten.  
DU SOLLTEST immer den vollen Pfad für Programme nutzen.

Beispiel : benutze C:Copy statt Copy in der Zeile.

In Kommandozeilen kannst du in ClassAction eingebaute ~Argumente~ und ~Kommandos~ benutzen.

- o Setze den RescanDir Zeiger, wenn du das aktuelle Verzeichnis wiedereinlesen lassen willst, nützlich, wenn die Aktion das Verzeichnis geändert hat.

### Knöpfe

- o Die beiden Pfeile nach oben und unten sind zum Sortieren der Aktionen gedacht.
  - o Der Knopf 'Laden' läßt dich ein Kommando auswählen.
  - o Der Knopf 'Kommandos' öffnet ein Auswahlfenster mit allen möglichen Kommandos und Argumenten für die Kommandozeile.
  - o Man kann eine existierende Aktion kopieren auf eine andere kopieren, indem man auf den Knopf rechts vom Namenfeld drückt und die entsprechende Klasse aussucht, dann doppelklickt und dann Doppelklick auf die entsprechende Aktion, und schon wird diese kopiert.
-

## 1.14 Startmodi

### CLI-Modus

Falls 'CLI' gewählt ist, wird die Aktion vom CLI gestartet und der eingegebene Stapelspeicher benutzt (vorgegeben ist 4096).

Ein CLI wird nur geöffnet, wenn's nötig ist (falls irgendetwas dargestellt werden muß).

Man kann eine Verzögerung für das CLI einstellen:

Wenn die Verzögerung negativ ist (d.h. Verzögerung = -1), wartet das CLI, bis du es mittels des Schließsymbols oben links schließt.

Wenn die Verzögerung null ist (d.h. Verzögerung = 0), schließt das CLI selbstständig nach Beendigung der Aktion.

Wenn die Verzögerung positiv ist (d.h. Verzögerung =n mit n>0), wartet das CLI n Sekunden, bevor es schließt, aber man auch das Schließsymbol benutzen.

Die Dimension des benutzten CLI kann im Tooltype CLISIZE eingestellt werden.

### WB-Modus

Falls 'WB' gewählt ist, dann wird kein CLI geöffnet und ClassAction simuliert einen Workbenchstart der Aktion.

Der erste String wird als Icon betrachtet, und die nächsten Argumente werden behandelt, als ob sie auf das Icon gezogen würden.

Gibst du '"[F]"' als Befehl an, dann wird der Dateiname gestartet als ob du doppelt auf das Icon geklickt hättest.

WARNUNG : dieser Modus ist nur mit Programmen, die Piktogramme besitzen, auszuführen, und nur echte Dateinamen (incl. Pfad) sollten angegeben werden.

### NO CLI-Modus

Falls 'No CLI' gewählt ist, wird kein CLI geöffnet, auch wenn das Programm etwas darstellen möchte, doch es läuft vom CLI.

### AREXX-Modus

Falls 'Arexx' gewählt ist, wird rx mit dem gegebenen Kommando ausgeführt. Das Kommando MUß ein Arexx-Skript sein.

Natürlich sollte RexxMaster aktiv sein und Rx sich im Verzeichnis Sys:rexxc/ befinden.

### INTERNAL-Modus

ClassAction hat einige eingebaute Befehle, die du aufrufen kannst.

---

Sie müssen in Großbuchstaben eingegeben werden. Variablen werden hier mit einem kleinen Buchstaben angegeben. n steht immer für die Nummer des Listers, 0 steht für die Quelle und 1 für das Ziel.

ADDICON	Füge ein Icon zur gewählten Datei hinzu. Es wird vom System abhängig von deinen DefIcons-Einstellungen ausgesucht
ALL	Wählt alle Dateien im Quellllister aus
COPY	Kopiert alle gewählten Dateien zum Ziel
COPYAS	Kopiert alle gewählten Dateien zum Ziel und gibt ihnen neue Namen
DELETE	Löscht alle gewählten Dateien
GETSIZE	Berechnet die Größe der gewählten Dateien und zeigt sie an
LIST n dir	Listet das Verzeichnis 'dir' im Lister 'n' dir: ein gültiger Pfad wie sys:tools, kann auch weggelassen werden, um alle Geräte anzuzeigen
MAKEDIR n	Erzeugt ein neues Verzeichnis im Lister 'n'. Ein Icon wird erzeugt, wenn die Option aktiv ist.
MOVE	Bewegt alle gewählten Dateien zum Ziel
MOVEAS	Kopiert alle gewählten Dateien zum Ziel und gibt ihnen neue Namen
NONE	Deselektiert alle Dateien im Quellllister
PARENT n	Geht ins übergeordnete Verzeichnis im Lister 'n'
PATTERN	Wählt alle Dateien, die einem Muster genügen. Du wirst nach einem Muster nach den Standardregeln gefragt (z.B. #? für mehrere unbekannte Buchstaben etc.)
REFRESH n	Liest Lister n erneut ein
SWAP	Tauscht den Inhalt von Quell- und Ziellister
TOGGLE	Wechselt für jede Datei von gewählt zu nicht gewählt und umgekehrt
TYPE	Wählt alle Dateien, die der selben Klasse wie die gewählte angehört. Dies kann einige Sekunden dauern, da jede Datei des Verzeichnisses untersucht werden muss.
WBINFO	Zeige WBInfo für alle gewählten Dateien. Du musst den Requester schließen, um weiterarbeiten zu können.

Anmerkungen:

- o Die Transfer-Befehle (COPY,COPYAS,MOVE,MOVEAS) prüfen, ob Dateien ersetzt würden und fragt dich, was du mit diesen Dateien tun willst. Dies wird nur auf oberster Ebene durchgeführt, also wirst du pro Verzeichnis nur einmal gefragt.
- o COPY funktioniert auch mit Archiven.
- o Einige Befehle können mit dem Stop-Knopf unten rechts unterbrochen werden. Er wird farbig, wenn dies möglich ist. Dann bleiben nicht bearbeitete Dateien angewählt.
- o Einige der Befehle können mit Cli-Befehlen nachgebaut werden, z.B. 'SURE[Wirklich löschen?]delete "[F]"' funktioniert fast genau so wie 'DELETE', aber die internen Befehle haben einige Vorteile, wie den Test auf Überschreibung und die

Fortschrittsanzeige.

SPACE-Modus

Dies ist kein echter Startmodus, es bedeutet nur, dass es sich nicht um einen Befehl, sondern um einen Abstand zwischen Icons handelt. SPACE kann nicht für Aktionen benutzt werden und ist sinnlos bei Tastaturbefehlen.

Siehe auch

Variablen

## 1.15 Variablen

Variablen

Zur Zeit sind 8 Argumente möglich :

die ersten 4 Kommandos sind kleingeschrieben : [f] [s] [b] [x] und sie enthalten das Resultat in Anführungszeichen

[f] : voller Pfad der gewählten Datei in Anführungszeichen  
[s] : voller Pfad der gewählten Datei ohne Endung in Anführungszeichen  
[b] : Dateiname in Anführungszeichen  
[x] : Dateiname ohne Endung in Anführungszeichen

die anderen 4 Kommandos tun dasselbe, bloß ohne Anführungszeichen

[F] : voller Pfad der gewählten Datei  
[S] : voller Pfad der gewählten Datei ohne Endung  
[B] : Dateiname  
[X] : Dateiname ohne Endung

Beispiel : sagen wir, du möchtest die Datei 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

---

Beispiel : sagen wir, du wählst die Datei ram:main.c

\* Die Kommandozeile

c:copy [f] [F].bak

wird ersetzt durch :

c:copy "ram:main.c" ram:main.c.bak

\* Die Kommandozeile

c:copy [f] [S].bak

wird ersetzt durch :

c:copy "ram:main.c" ram:main.bak

## 1.16 Kommandos

Zur Zeit sind 5 Abfragekommandos verfügbar :

REQD[Text] : Fragt nach einem Verzeichnis.

REQF[Text] : Fragt nach einer Datei.

REQV[Text] : Fragt nach einem Laufwerk.

REQT[Text] : Fragt nach einem Text.

SURE[Text] : Fragt nach der Zustimmung des Nutzers.

=====

REQ Kommandos

REQ Kommandos öffnen ein Reqtoolsrequester mit einem kleinen Titel [Text].

Dies könnte man in interaktiven Kommandozeilen benötigen.

Beispiel :

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

Dies wird ein Auswahlfenster öffnen, um das Zielverzeichnis zu bestimmen, und die gewählte Datei [f] wird dann in dieses entpackt.

REQF[] macht dasselbe, außer daß es nach einer Datei fragt.

Beispiel :

c:dir [f] > REQF[Wähle eine Datei]

REQV[] fragt nach einem Laufwerk.

REQT[] fragt nach einem Text, falls man Argumente benötigt, zum Beispiel :

c:cpu REQT[Gib Argumente für CPU an]



=====

SURE Kommando:

Das SURE[Text] Kommando öffnet ein Requester mit dem Text [Text] und zwei Knöpfen : yes / no.  
Wenn man no wählt, wird die Kommandozeile abgebrochen.  
Wenn man yes wählt, wird ClassAction die Kommandozeile im RECHTEN Teil des Sure-Kommandos ausführen.

Beispiel :

SURE[Diese Datei wirklich löschen ?]C:delete [f]

Es wird ein requester geöffnet, das den Nutzer fragt "Diese Datei wirklich löschen ?".  
Wenn der Nutzer mit no antwortet, passiert nichts; wenn er aber mit yes antwortet, dann wird C:delete [f] ausgeführt.

=====

Natürlich kann man jede Anzahl von Argumenten / Kommandos in einer Kommandozeile kombinieren.

Beispiel :

SURE[Diese Datei wirklich umbenennen]c:rename [f] REQF[Nenne den neuen Namen]

## 1.17 Zukünftige Verbesserungen

Nichts im Moment ....

## 1.18 Über

Aktuelles ClassAction-Team:

Planung, Entwicklung, Koordination:

Martin R. Elsner

Grafik:

Martin Eriksson (der andere Martin ;)

Beta-Tester:

Christian Aichinger

Jan Britsch

Jean-Pierre Riviere

Jimmy Sulter  
Jocke Sjöblom  
Martin Eriksson  
Patric Hofmann  
Rolf Max Rotvel  
Stefan Funke

Übersetzer:

Abdul Alkaç (Holländisch)  
Arturo Franzin (Italienisch)  
Bigo (Französisch)  
Javier de las Rivas (Spanisch)  
Martin R. Elsner (Deutsch)  
Márton Dósa (der ungarische Martin ;) (Ungarisch)  
Sinan Gürkan (Türkisch)

Ich möchte den folgenden Leuten für ihren Beitrag zur Wiedergeburt von ClassAction danken :

- o allen Usern, die mir Emails geschickt und ihre Meinung mitgeteilt haben (sei nicht traurig, wenn ich deinen Namen hier nicht aufgeführt habe ;)
- o Evan Scott und Thies Wellpott für ihre Erlaubnis, FTPMount zu benutzen und weiterzuentwickeln
- o Mick Saile für die Erlaubnis, seine GlowIcons zu benutzen (opusgibar, erhältlich in Aminet)
- o Dietmar Eilert für das großartige GoldED, das ich beim Programmieren benutze
- o und Salim Gasmi für die Entwicklung der Versionen 1.0 bis 3.6 dieser tollen Software!

Martin

Ich möchte den folgenden Leuten danken :

- o Philippe Thomas (für Vorschläge, Hilfe, Betatesten, das französische Guide und UTT, welches für die Installation benutzt wurde)

Hey Phil, if you read this, click ~here~

- o Bruno Durremberger (für's Betatesten)
  - o Jean Michel Dessolas (für's Betatesten auf einem A4000/40)
  - o Joern Krueger für's Betatesten und die guten Ideen.
  - o Obvious Implementations Corp (für Dice C Pro)
  - o Nico Francois für die ReqTools.library
-

- o Georg Hörmann für die \*GROßARTIGE\* xfdmaster.library
- o Stefan Stuntz für das wundervolle Magic User Interface .
- o Allen Nutzern, die mir Fehlermeldungen und Vorschläge schickten

Salim

## 1.19 Der Autor

Du kannst Martin unter folgender Adresse erreichen :

Martin R. Elsner  
Müsener Straße 46

57223 Kreuztal  
Germany

Web: <http://www.martin-elsner.de>

E-Mail: [classaction@martin-elsner.de](mailto:classaction@martin-elsner.de)

oder den Koautor Salim :

Gasmi Salim  
4b rue des petits champs

67300 Shiltigheim  
France

Irc: Dr\_Unix (#amigafr, #amiga)

Internet: <http://www.gasmi.net>

E-Mail: [salim@gasmi.net](mailto:salim@gasmi.net) oder [salim@sdv.fr](mailto:salim@sdv.fr)

## 1.20 Benutzung von ClassAction

Benutzung von ClassAction

Die Benutzung von ClassAction ist WIRKLICH einfach. Wenn du nicht weißt was wozu dient, bewege die Maus über das entsprechende Icon und schau dir den Hinweis am unteren Rand an (du musst dafür die Option ShowHelp aktiviert haben).

Wähle eine Datei im Auswahlfenster. Du kannst mit dem entsprechenden Knopf ins Mutterverzeichnis springen.

---

In dieser Liste kannst du dich wie in einem normalen Dateiauswahlfenster bewegen. Links vom Pfad befindet sich ein Pfeil, der dich ins übergeordnete Verzeichnis bringt.

Rechts vom Pfad gibt es zwei Popup-Menüs: das erste zeigt eine Liste vordefinierter Pfade, die auch geändert werden können; das andere zeigt die Verzeichnisse an, die du schon besucht hast, es ist die Fenster-Geschichte. Wähle einfach einen der Pfade, um das Verzeichnis aufzulisten.

Besondere Aktionen sind mit den Icons oben und unten möglich.

Manche Methoden können über Tasten aufgerufen werden; die Tasten sind in ClassActionPrefs definiert und können angeschaut und geändert werden.

Bei Standardeinstellung kann man z.B. immer die ausgewählten Dateien mit Hilfe von DEL löschen, und HELP zeigt diese Anleitung an.

Nach der Auswahl sieht man im rechten Fenster die Klasse der Datei und die möglichen Aktionen. Nun wähle die gewünschte Aktion...

Drücke eine Taste, und die erste Datei, die mit diesem Buchstaben beginnt, wird angezeigt. Drücke zusätzlich SHIFT, und das erste Verzeichnis wird angezeigt.

(Dies kann von deinen eigenen Befehlen überschrieben werden, die z.B. die ausgewählten Dateien durch Drücken von "c" kopiert.)

Wenn man einen Doppelklick auf einer Datei ausführt, wird die erste definierte Aktion ausgelöst.

Das Fenster von ClassAction ist ein AppWindow, man kann also Piktogramme raufwerfen.

Du kannst das Einstellungsprogramm starten, indem du 'Einstellungen' im Menü auswählst.

Verlassen kannst du das Programm, indem du auf das Schließsymbol klickst oder 'Quit' im Menü auswählst.

Um das Fenster in ein AppIcon zu verwandeln, einfach auf das Symbol klicken oder 'Verbergen' im Menü auswählen.

Ein Doppelklick auf das AppIcon öffnet das Fenster wieder.

Wenn ClassAction ein AppIcon ist, kann man Piktogramme raufwerfen und der Dateityp wird erkannt.

Ein Fenster mit den möglichen Aktionen öffnet sich, so daß man eine auswählen kann.

Die Benutzung des Dateimanagers ist WIRKLICH einfach, das LINKE Fenster ist IMMER das Quellverzeichnis und das RECHTE IMMER das Zielverzeichnis.

Siehe auch

ARexx Befehle

Knöpfe

Konfiguration

Menü

## 1.21 Die Knöpfe

Dies ist eine Liste der vordefinierten Befehle. Du kannst sie ändern und neue erzeugen, mit Hilfe von `ClassActionPrefs`.

### Auswahlknöpfe

Über der Quellliste befinden sich 6 Knöpfe, die die Auswahl erleichtern:

All: Wähle alle Einträge im Lister, incl. Verzeichnisse.  
None: Wähle alle Einträge ab.  
Pattern: Wähle alle Dateien, die in das von dir angegebene Muster passen.  
Verzeichnisse werden nicht ausgewählt.  
Type: Wähle alle Dateien der selben Klasse wie die ausgewählte.  
Verzeichnisse werden nicht ausgewählt.  
Toggle: Wechsele (nicht) ausgewählte Einträge.  
Wirkt sich auch auf Verzeichnisse aus.  
GetSize: Berechnet und zeigt die Größe aller markierten Dateien und Verzeichnisse.  
Swap: Tausche den Inhalt von Quell- und Zielliste.

### Verzeichnisknöpfe

Für beide Listen existieren Knöpfe, um sich darin zu bewegen und neue Verzeichnisse zu erzeugen.

Parent: Zeige das übergeordnete Verzeichnis.  
Wenn du im Basisverzeichnis bist, wird die Geräteliste gezeigt.  
Devices: Zeige Laufwerke und Assigns.  
MakeDir: Erzeuge ein neues Verzeichnis; du wirst nach einem Namen gefragt.  
Wenn die Option 'CreateIcons' aktiv ist, wird auch ein Icon erzeugt.

### Aktionsknöpfe

Manche Aktionen können mit diesen Knöpfen auf die gewählten Dateien angewandt werden:

WbInfo: Zeige die Workbench-Icon-Information der ausgewählten Dateien.  
Dort kannst du Tooltypes, Schutz-Bits, ... ändern.  
AddIcon: Erzeuge ein Icon für die ausgewählten Dateien.  
Wenn es schon ein Icon gibt, wirst du informiert, und nichts passiert.  
Rename: Gib den ausgewählten Dateien neue Namen.  
Delete: Lösche die gewählten Dateien. Du wirst gefragt, ob du dies wirklich willst.

### Transfer-Knöpfe

Du kannst Dateien mit Hilfe folgender Knöpfe bewegen:

Copy: Kopiere die gewählten Dateien zur Zielliste.  
CopyAs: Kopiere die gewählten Dateien zur Zielliste, und gib ihnen einen neuen Namen.  
Move: Verschiebe die gewählten Dateien zur Zielliste.  
MoveAs: Verschiebe die gewählten Dateien zur Zielliste, und gib ihnen einen neuen Namen.

Andere Knöpfe

Es kann weitere Knöpfe geben, um die Möglichkeiten zu zeigen.  
Schau in den Einstellungen nach, um Informationen über die Funktion dieser Knöpfe zu erhalten.

Siehe auch

Befehle  
ClassActionPrefs

## 1.22 Das Menü

Projekt

Hilfe: Zeige diese Anleitung.  
Alternativ kannst du auch die 'HELP'-Taste drücken.  
Einstellungen: Starte ClassActionPrefs, um die Einstellungen anzupassen.  
Über: Zeige Informationen über das Programm.  
Verbergen: Schließe das Fenster und erzeuge ein AppIcon.  
Verlassen: Verlasse das Programm.

Einstellungen

Fenstergröße merken: Speichere die aktuelle Fensterposition und -größe.  
Wenn ClassAction das nächste Mal startet, wird es diese Dimensionen benutzen. Sie können in ClassActionPrefs auch exakte Werte angeben.

Die folgenden Optionen sind identisch zu den Optionen mit den gleichen Namen.  
Wenn du also Optionen dauerhaft ändern willst, dann musst du die Einstellungen ändern.

## 1.23 Technische Infos

Technische Informationen

---

ClassAction wurde zuerst mit DICE C 3.0 geschrieben und wird nun in C++ mit StormC 3.0 weiterentwickelt, unter Verwendung von ReAction (toller Name ;-).

Es ist echtes C++ mit einigen schönen Klassen für ARexx, das Menü, die Einstellungsdateien ... Wende dich an den Autor, wenn du dir den Quelltext anschauen möchtest.

Weitere Infos :

Die Einstellungsdatei ist eine ASCII-Datei namens  
ENVARC:ClassAction.prefs

DISK libraries, die benutzt werden, falls vorhanden :

```
xadmaster.library
xfdmaster.library      V30+
```

Wie erkennt ClassAction eine Klasse :

- 1- Prüfung, ob der Dateiname mit einem definierten Erkennungsnamen übereinstimmt.
- 2- Prüfung, ob die Offsets mit denen definierter Klassen übereinstimmen.
- 3- Entpacken der Datei mittels der xfdmaster.library.
- 4- Prüfung, ob der entpackte Puffer übereinstimmende Offsets mit definierten Klassen hat.
- 5- Prüfung, ob die Datei eine ASCII-Datei ist (wenn das ASCII Offset Kommando besteht)

Wenn alles fehlschlägt, wird die Datei der  
'Unbekannten Klasse' zugeordnet.

## 1.24 Installation

Installation

Benutze einfach das Installationsskript in diesem Archiv.  
Klicke dazu auf das entsprechende Piktogramm.

Wenn du nach der Installation Dateien bewegst, musst du die Einstellungen ändern. Starte einfach ClassActionPrefs und überprüfe die Pfade (Anleitung, ClassActionPrefs, Icons). Ändere sie und speichere. Sobald ClassAction das nächste Mal startet, wird es wissen, was passiert ist ;)

Siehe auch

Nach der Installation

---

## 1.25 Rechtliches

### Copyright

ClassAction und ClassActionPrefs sind Copyright  
© 2001 Martin R. Elsner und Salim Gasmi.

ClassAction ist eine Mailware Programm. Das Paket darf in keiner Weise verändert werden und nicht für kommerzielle Zwecke benutzt werden ohne Zustimmung des Autors. Die Copyright Mitteilung ist geschützt.

### Garantie

Für Schäden durch Benutzung dieses Programms wird keinerlei Haftung übernommen. Die Benutzung erfolgt auf eigenes Risiko. Die Software wird "so wie sie ist" bereitgestellt, ohne jegliche Garantieerklärungen oder anderweitige Versicherungen auf Lauffähigkeit und Stabilität der Software und Dokumentation. Die Dokumentation ist nach bestem Wissen und Gewissen erstellt, aber der Autor behält sich das Recht vor, die Software und/oder Dokumentation ohne Bemerkung zu erneuern.

## 1.26 Geschichte

### Geschichte

22.04.01 : V4.1

- Beide Lister zeigen Name, Größe, Datum und Schutzbits, und können nach Name, Größe oder Datum sortiert werden (einfach auf den Spaltentitel klicken). Spalten können durch Anklicken des "unsichtbaren" Trennstrichs angepasst werden.
- LHA- und ZIP-Archive können wie wirkliche Verzeichnisse durchsucht werden!
- Das PopUp-Fenster enthält nun Buttons, POPUPWIDTH und POPUPHEIGHT sind überflüssig, das Fenster wird immer angezeigt.
- Tooltype DATESTRING hinzugefügt, um "heute" usw. statt dd-mmm-yy anzuzeigen.
- Tooltype MUIQUIT hinzugefügt, um das Fenster mit dem Schließen-Gadget zu iconifizieren.
- Tasten KEY\_DEL (standard: DEL) und KEY\_RENAME (F4) als Abkürzung für Löschen und Umbenennen hinzugefügt.
- "Tauschen" neu geschrieben und viel schneller.
- "Kopieren" benutzt nun die Option CLONE, um Datum und Schutzbits zu erhalten.
- Ersetzen-Requester zeigt nun Größe und Datum der zwei Dateien.
- Text-Requester wird immer unter der Maus geöffnet.
- Nach Aktionen wird wieder die alte Position in der Liste



gezeigt, und nicht der Anfang.

- Busy-Pointer und Dateinamen-Anzeige w&auml;hrend Kopieren,...
- Pfad Buttons haben immer die selbe Gr&ouml;&szlig;e und werden wirklich entfernt, wenn man weniger als 13 Pfade angibt.
- Fehler behoben: .info kopieren funktioniert richtig, "bewegen als" auch, WBInfo zeigt nun Datentr&ouml;ger korrekt (Dank an Stephan Rupprecht!)

31.03.01 : V4.0

- Das ganze Program "neu geschrieben" um Reaction und neue OS3.5+ -Features zu nutzen.
- Jetzt k&ouml;nnen nur noch 13 Verzeichniskn&ouml;pfe konfiguriert werden (sollte ausreichen!).
- Manager-Actionen sind jetzt Bilder!
- Ger&atilde;t&euml; werden jetzt in der normalen Dateiliste angezeigt.
- Neue Kn&ouml;pfe: MakeDir auch f&uor; die Zielliste, WbInfo, AddIcon, MoveAs, alle Selektionskn&ouml;pfe.
- Men&uuml; hinzugef&uuml;gt(!) mit Hilfe, Einstellungen, ...
- Asynchrone Hilfe hinzugef&uuml;gt!
- Fortschrittsanzeige f&uor; kopieren/verschieben ... hinzugef&uuml;gt.
- Jetzt ist die Klasse einer Datei diejenige mit den MEISTEN passenden Offsets, nicht die erste, die passt.
- Kommentar-Optimierung und Tooltype NOCOMMENT entfernt.
- Neue Tooltypes: COPYICONS,CREATEICONS,DRAWERSFIRST,HELPPFILE,SHOWHELP,X,Y,WIDTH,HEIGHT,POPUPWIDTH,POPUPHEIGHT, mehr Tasten.
- ARexx-Port ist jetzt CLASSACTION.1 um die Handhabung zu erleichtern.

28/03/98 : V3.6

- Ein neues Programm namens CStart, welches als Projekt f&uor; deine Piktogramme benutzt werden kann, hinzugef&uuml;gt
- ClassAction setzt den Dateityp nun ins Kommentar, um die Erkennung zu beschleunigen.
- Tooltype NOCOMMENT hinzugef&uuml;gt, zum Deaktivieren der Dateikommentare.
- Der Hauptverzeichnislister von ClassAction ist jetzt auch tastaturgesteuert, mit den Pfeiltasten/Leertaste/Enter.
- Neu : wenn du eine Taste dr&uuml;ckst, wird die erste Datei, welche mit diesem Buchstaben beginnt, ausgew&ahlt (Shift+Taste f&uor; Verzeichnisse)
- Taste zur Musterauswahl hinzugef&uuml;gt (einstellbar &uuml;ber Tooltype)
- Tasten zur Dateiauswahl hinzugef&uuml;gt [alle anw&ahlen, keine anw&ahlen, Auswahl tauschen] (einstellbar &uuml;ber Tooltypes)
- Taste f&uor; Mutterverzeichnis hinzugef&uuml;gt (einstellbar &uuml;ber Tooltype)
- Fehler in MoveDir entfernt
- Fehler im "Verzeichnis erstellen"-Auswahlfenster entfernt
- Einige Routinen optimiert und beschleunigt.
- Das Abst&urzen und H&ang;enbleiben von Kommandos in ClassActionPrefs behoben
- Einige andere kleine Fehler behoben

27.06.97 : V3.5

- Bonus Version.
- CA\_Register wird nicht mehr gebraucht, Registration kann On-Line auf meiner Seite <http://www.gasmi.net> gemacht werden.

- Dateimanager behandelt nun Piktogramme
- Tooltype DISPINFO hinzugefügt
- Tooltype GTLIKE verändert
- Kleinere Fehler entfernt

07.02.97 : V3.4

- Dies ist wahrscheinlich die LETZTE Veröffentlichung von ClassAction
- Einige kleinere Fehler berichtigt
- Tooltype ICONSTART für die Ikonifizierung von ClassAction beim Start hinzugefügt, für die Leute, die nicht wissen, wie man das im MUI Voreinsteller macht

23.08.96 : V3.3

- ClassAction ist keine ShareWare mehr, dafür MailWare !!!!
- Ca\_Register für freie Registration eingeführt
- Einige Fehler beseitigt.

07.07.96 : V3.2

- ClassAction wurde mit der neuesten Version von Dice neu kompiliert und ist stabiler.
- Arexx-Unterstützung ist nun vollständig.
- Einige Fehler beseitigt.

09.04.96 : V3.1

- ClassAction besitzt nun einen kleinen eingebauten Dateimanager.
- Einige Fehler beseitigt.

12.03.96 : V3.0 (Großes Update)

- ClassAction und ClassActionPrefs nutzen nun MUI 3.0+
- ClassAction kann bis zu 50 Pfadknöpfe haben
- ClassAction erkennt Verzeichnisse.
- ClassAction kennt eine Mehrfachauswahl.
- ClassAction hat ein Appwindow.
- Man kann jetzt die Lerngenauigkeit einstellen.
- Tooltypes CAPREFS,GTLIKE hinzugefügt.
- Tooltypes HEIGHT,APPSTART,WINX,WINY,ICONX,ICONY,CX\_PRIORITY,CX\_HOTKEY,REQBUG,STARTDIR,WBFONT,PUBSCREEN entfernt, da nicht mehr vonnöten.

25.09.95 : V2.8

- ClassAction hat nun VOLLE Commodity Unterstützung und einen Hotkey.
- Tooltype APPSTART kann auf HIDE (APPSTART=HIDE) gesetzt werden, wenn ClassAction verdeckt gestartet werden soll.
- Tooltype CX\_HOTKEY hinzugefügt
- Tooltype PUBSCREEN hinzugefügt für Arbeit auf öffentlichen Bildschirmen
- Die Interne Dateiauswahl von ClassAction gibt nun wieder allen Speicher frei.

11.09.95 : V2.75

- Die REQD,REQF,REQV Kommandos waren nicht inkompatibel mit dem AppIcon Modus, aber mit einigen Programmen, wie MagicMenu (So ein Pech, ich benutze MagicMenu...)  
Viele Nutzer meldeten einen automatischen Sprung ins REQT Kommando, sogar ohne inkompatible Programme zu benutzen.

Ich fügte einen Tooltype (REQBUG) hinzu, damit der Nutzer den Auto-Sprung in REQT überwachen kann.

03.09.95 : V2.7

- Kommandos [b],[x],[B],[X],[F],[S] hinzugefügt.
- Das Aktion Arexx Kommando hinzugefügt.
- Einige Fehler behoben
- Einen Fehler behoben, wenn mehrere Piktogramme auf's AppIcon geworfen werden.
- Die REQD,REQF,REQV Kommandos springen nun in REQT, wenn sie vom AppIcon oder im Arexx-Modus benutzt werden, sie sind nur mit dem Fenstermoduskompatibel.

28.08.95 : V2.6

- Das Fenster von ClassAction ist nun in der Größe veränderbar.
- ClassAction und ClassActionPrefs nutzen nun die ReqTools.library.
- REQV Kommando zur Frage nach einem Laufwerk hinzugefügt.
- REQT Kommando zur Frage nach einem Text hinzugefügt.
- WINX und WINY Tooltypes hinzugefügt.
- Das Lernrequester hat jetzt einen ALLES Knopf.
- Die gewählte datei ist jetzt in allen REQ Requestern.
- ClassAction nutzt nun für registrierte Nutzer einen Keyfile in S:.

17.07.95 : V2.5

- ClassAction und ClassActionPrefs sind jetzt lokalisiert und ein französischer Katalog ist beigelegt.
- Die Lernfunktion ist zugefügt.
- Um die erste Aktion einer datei zu starten, reicht nun ein Doppelklick.
- Allgemeine Aktionen sind nun synchron und lesen das aktuelle Verzeichnis wieder ein.
- AUTOSELECT ToolType wird nicht mehr benutzt.
- REQ Requester öffnen jetzt das aktuelle Verzeichnis.
- Kleinere Verbesserungen gemacht.

12.06.95 : V2.1

- Die eingebaute Klasse 'Allgemeine Aktionen' hinzugefügt.
- SURE[] Kommando hinzugefügt.
- ASCII[] Offset Kommando zum Erkennen von ASCII Dateien hinzugefügt.
- Arexx Kommandos hinzugefügt: AppIconify, Show, Status, GetClass.
- Tasksystem geändert, jetzt nutze ich Systemtags(). Wir brauchen keine tmp Dateien mehr.
- Man kann nun im CLI Modus eine Verzögerung definieren.
- 'string' und "string" für Offsetdefinition hinzugefügt.
- ToolType OUTPUT ist nun überflüssig und wird nicht mehr benutzt, man kann dafür den Tooltype CLISIZE nutzen.
- Aktionsrequester sind nun richtig proportioniert, erscheinen unter'm Mauszeiger und nutzen den vorderen Bildschirm.
- Knöpfe 'Benutzen' und 'Speichern' vertauscht und 'Abbruch' hinzugefügt in ClassActionPrefs, um dem Amigaaussehen zu folgen, den Knopf 'Über' als '?' in die rechte obere Ecke verschoben.

- ClassActionPrefs Cycle-Gadgets waren nicht 100% systemkonform und einige Patches, wie CyclettoMenu verursachten Fehler mit ClassActionPrefs; das ist nun behoben.
- Erkennungscode und Inforoutine verbessert: diese sind nun bis zu 400% schneller.
- Farbfehler bei Highlight entfernt.
- ClassAction schließt nun beim Ikonifizieren den WB-Screen nicht mehr.
- Einen gemeinen Fehler in ClassActionPrefs gefunden und beseitigt.
- Die rechte Maustaste zeigt Zuweisungen nur, wenn die Maus im Requester ist und rechte Maustaste bringt auch das Verzeichnis zurück.
- ClassAction speichert nun die Fensterposition.

23.05.95 : V2.00 ( Großes Update )

- ClassAction hat jetzt ein AppIcon.
- ClassAction ist jetzt ein Commodity.
- ClassAction hat jetzt einen Arexx Port.
- ClassAction nutzt jetzt den eingestellten WB-Zeichensatz.
- Exec Modus 'Arexx' hinzugefügt.
- Verschiedene Farben für Verzeichnisse/Dateien.
- Hoch/Runter Symbole in ClassActionPrefs.
- 'Benutzen' Knopf in ClassActionPrefs.
- Klassen werden nun in ClassActionPrefs sortiert.
- APPSTART, ICONNAME, ICONX, ICONY, CX\_PRIORITY, WBFONT, OUTPUT, ICONFILE ToolTypes hinzugefügt.
- Wenn das Fenster ikonifiziert ist, hat es die richtige Höhe für den eingestellten Zeichensatz.
- Neues Speicherformat (CASF20).
- Endungs/Vorsilben Knopf entfernt. Ersetzt durch Erkennungsnamen Knopf, der alle Wildcards akzeptiert.
- Konfigurationsdatei in ENVARC: verschoben.
- einen Installer hinzugefügt.
- Code optimiert.

05.05.95 : V1.43

- Requestercode bereinigt, jetzt 5% schneller.
- ClassAction schaut jetzt nach seinem Namen in der WBStartup-Struktur und kann somit umbenannt werden.

02.05.95 : V1.42

- 'No Cli' Exec Modus in ClassActionPrefs eingefügt.
- "" werden immer bei Dateinamen benutzt, das ist einfacher für ARexx-Skripts.

06.04.95 : V 1.4

- ToolType HEIGHT hinzugefügt.
- Code optimiert.

22.02.95 : V 1.31

- Bei Doppelklick auf dieselbe Datei wird die erste Aktion ausgeführt (ist schneller, als erst die Aktion zu wählen).
- diese Version ist jetzt ShareWare und man muß sich registrieren lassen.

15.01.95 : V 1.3

---

- REQD[] und REQF[] als interaktive Kommandos hinzugefügt.
- Knopf 'Kopieren' in ClassActionPrefs eingefügt.
- kleinere Verbesserungen gemacht.
- Betatester berichten, daß diese Version sehr stabil läuft.

25.11.94 : V 1.22

- Erste Veröffentlichung
- Knopf 'Info' hinzugefügt.
- ClassAction hatte Locking() die Verzeichnisse, wenn es sie mit UnLocking() gelesen hat .... \*BEREINIGT\*
- Codeoptimierung
- Kleinere Fehler entfernt.

08.11.94 : V 1.21

- Programm stürzte mit leeren Diskettenlaufwerken ab .. \*BEHOBEN\*
- Laufwerks/Namen Fehler behoben
- <..> entfernt im Hauptverzeichnis.

07.11.94 : V 1.2

- Neue Schnittstelle, Ich habe meine eigenen Dateiauswahlfenster.
- STARTDIR & DRIVE1 to DRIVE11 ToolTypes hinzugefügt.

01.11.94 : V 1.1

- Benutzt jetzt die xfdmaster library zum Erkennen und Entpacken von Dateien.
- Konfiguration mit ToolTypes hinzugefügt (DECRUNCH,AUTOSELECT).
- 'Unbekannte Klasse' ist jetzt eine eingebaute Klasse mit unbegrenzter Anzahl von Aktionen.
- Neues Speicherformat (CASF11).

16.10.94 : V 1.0

- Neues Speicherformat (CASF10).
- Fenster hat jetzt ein Größensymbol.
- eine Menge Klassendefinitionen hinzugefügt.

10.10.94 : Beta Version

- tmp Dateifehler entfernt.
- Offset Zählfehler behoben.
- benutzt asl.library für die Dateiauswahlfenster.

01.10.94 : Alpha Version

## 1.27 Die Arexx Kommandos

ARexx Befehle

ClassAction hat einen Arexx Port namens : CLASSACTION.1

Hier ist eine Liste der Arexx Kommandos :

=====

Quit

---

Beendet ClassAction...

Use

Zwingt ClassAction, die Einstellungsdatei erneut zu lesen.

Ver

Gibt die Version von ClassAction zurück.

Status

Gibt den aktuellen Staus von ClassAction zurück.

Return 0 : ClassAction ist Ikonifiziert.

Return 1 : ClassAction ist im Fenstermodus.

AppIconify

ZwingtClassAction in den Ikonifizierungsmodus.

Wenn ClassAction schon ikonifiziert ist, tut dieses Kommando gar nichts.

Show

ClassAction zeigt das Hauptfenster.

Wenn ClassAction schon im Fenstermodus ist, tut das Kommando nichts.

Load <Dateiname>

ClassAction versucht die Datei <Dateiname> zu laden, und öffnet ein Aktionsfenster zur Auswahl.

Wenn die Datei nicht existiert, tut das Kommando nichts.

GetClass <Dateiname>

ClassAction versucht die Datei <Dateiname> zu laden und ihre Klasse zurückzugeben.

Wenn die Datei nicht existiert, tut das Kommando nichts.

=====

Action <Dateiname> <Aktionsmuster>

ClassAction führt die erste zutreffende Aktion für die Datei <Dateiname> aus.

Beispiel : Action ram:toto.lha extr

startet die erste Aktion, deren Name 'extr' enthält mit der Datei ram:toto.lha

Das Muster ist nützlich, damit man nicht den exakten Namen der Aktion angeben muß.

Bemerkung: Schau ins Rexx/ Verzeichnis des Archivs, um ARexx-Skripts zu finden.

## 1.28 Die Archiv-Unterstützung

Archiv-Unterstützung

ClassAction unterstützt einige spezielle Funktionen für Archive. Wenn du doppelt auf eine Archivdatei klickst, wird dies wie ein wirkliches Verzeichnis geöffnet, und sein Inhalt wird angezeigt. Du kannst dich darin bewegen, Dateien aus einem Archiv auf ein Laufwerk oder von einem Laufwerk in ein Archiv kopieren und sogar definierte Aktionen ausführen. Andere Aktionen sind immer noch über die Aktionen-Liste möglich, z.B. zum Entpacken des gesamten Archivs. Die Dateierkennung basiert auf den Endungen, um eine schnelle Erkennung zu gewährleisten. Nenne also keine Dateien \*.lha, die keine Archive sind!

Übersicht

Unterstützte Dateitypen:

- o \*.lha Dateien (bzw. \*.lzx / \*.lzh)
- o \*.zip Dateien

Unterstützte Aktionen:

- o Ein Doppelklick auf eine archivierte Datei öffnet ein Popup-Menü mit den klassenspezifischen Aktionen
- o COPY (interner Befehl) in ein Archiv / aus einem Archiv (NICHT von einem zu einem anderen!)

Technische Informationen:

- o Lesen und Dateien entpacken wird mit `xadmaster.library` realisiert.
- o Das Archivieren von Dateien (also Kopieren in ein Archiv) benötigt die Befehle `"lha"` und `"zip"` irgendwo im Pfad deines Systems.
- o Zum Ausführen von Aktionen wird die Datei in `t:` gespeichert.

## 1.29 FTPMount

FTPMount

Was ist FTPMount ?

Nun, FTPMount kann Ihnen helfen, Dateien mit FTP zu übertragen. Der Unterschied zu gut bekannten FTP-Programmen besteht in der Methode, die Verbindung und den Transfer zu realisieren:

Eine Verbindung wird wie ein normales Verzeichnis gehandhabt. Das bedeutet, dass man Standardprogramme oder die Shell anstelle von FTP-Befehlen benutzen kann.

Dies ist der Grund dafür, dass FTPMount eine großartige Erweiterung zu ClassAction darstellt: ClassAction wird ein FTP-Programm, und man muss nicht zwischen mehreren Dateimanagern wechseln.

Installation von FTPMount

Zunächst musst du dir FTPMount besorgen. Versuch es einfach auf der ClassAction Support-Side, dort gibt es immer die aktuelle Version.

Entpacke das Archiv (mit ClassAction ...) und starte das Installationsprogram.

Die besten Einstellungen sind:

Geübter Benutzer

Kopiere DOSDriver-Mount-Eintrag nach: `sys:wbstartup`

Der Rest hängt von deinem System ab, also wähle die passenden Optionen.

Dies hat nur die benötigten Dateien kopiert und die nötigen Assigns angelegt. Du musst FTPMount nun konfigurieren:

Starte ClassAction und geh in das Verzeichnis, in dem FTPMount gespeichert ist (nicht über `FTPMountDir:`, du musst den wirklichen Pfad benutzen, z.B. `work:ftpmount`); dann weiter ins Verzeichnis `"Hosts"`.



Dort findest du einige Verzeichnisse. Erzeuge ein neues (mit Icon!), nenne es nach der Verbindung, die du hinzufügen möchtest (der Name hat keine weitere Bedeutung).

Dann klicke auf das Verzeichnis und wähle WBInfo. Hier musst du ein paar Tooltypes hinzufügen:

Mindestens

HOST=..

z.B. HOST=ftp.uni-paderborn.de wenn du eine Verbindung mit dieser Seite wünschst. Abhängig von der Seite musst du auch den Benutzernamen und das Passwort angeben:

USER=...

PASSWORD=...

Außerdem kannst du mit

ROOT=...

den Pfad festlegen, der zuerst angezeigt wird, z.B.

ROOT=/pub/aminet

Wenn du während der Verbindung ein Status-Fenster sehen möchtest, füge noch

STATUS

hinzu. Weitere Optionen findest du in der Anleitung von FTPMount.

Nun kannst du die Tooltypes speichern, und die Konfiguration ist (für diese Verbindung!) beendet.

FTPMount nutzen

Wenn du FTPMount erfolgreich installiert und konfiguriert hast, könnte es nützlich sein, den Rechner neu zu starten (abhängig vom Ort, an dem der DOS-Treiber installiert worden ist; du kannst FTPMount auch mit einem Doppelklick auf den Treiber starten, das sollte ausreichen).

Dann kann das Spiel beginnen:

Starte ClassAction.

Starte ein TCP-Programm wie Genesis oder Miami, um den AmiTCP-Port zu starten, geh online.

Dann geh in ClassAction ins Verzeichnis FTPMount:

Wähle das Verzeichnis, das du eingefügt und konfiguriert hast.

Wenn alles richtig installiert ist, wird ein Fenster erscheinen, das anzeigt, dass FTPMount versucht, die Verbindung herzustellen. Wenn dies gelingt, wird der Inhalt der FTP-Seite im ClassAction-

Lister angezeigt.

Benutze Methoden wie kopieren, löschen, ... um Daten zu bewegen. Manche Methoden sind nicht möglich (vor allem die Klassenerkennung mit Hilfe von Offsets).

Wenn du die Option "STATUS" angegeben hast, wirst du immer Informationen über die Aktionen von FTPMount sehen.

Normalerweise wird die Verbindung nach einigen Minuten Untätigkeit geschlossen, kann aber wieder geöffnet werden, solange du online bist.

Wenn du alles erledigt hast, geh offline, das ist alles, FTPMount wird alle Verbindungen schließen und du kannst mit anderen Dingen fortfahren!

## 1.30 Was sollte man nach der Installation tun

Was nun ?

Nach der Installation solltest du ein funktionierendes ClassAction besitzen, mit einigen Klassendefinitionen, aber sehr wenigen Aktionen (viele Klassen haben nicht mal welche).

Es liegt an dir, Aktionen zu definieren für dein System und deine Programme.

Lade ClassActionPrefs und konfiguriere nach deinen Wünschen...

Wenn du nicht weißt wie, dann lies dieses Guide :)).

Nach der Konfiguration der Klassen und Aktionen konfiguriere ClassAction mit seinen Tooltypes.

Es mag lange dauern, bis eine 'feine' Konfiguration erstellt ist. Aber wenn sie da ist, ist's GROßARTIG !!!

Ein kleiner Tip: benutze nicht zu viele Klassen:  
je mehr man definiert, desto länger wird ClassAction brauchen, um die Klasse zu finden, zu der die Datei gehört. Entferne einfach alle Klassen, die du nicht benutzt.

Siehe auch

Klassen  
Konfiguration

## 1.31 Mitteilung für 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... :)).

## 1.32 Bekannte Fehler

Fehler ??

Es können Probleme auftauchen, wenn DirectoryOpus installiert ist, da dann nicht alle Funktionen der Workbench aktiv sind. Wenn du solch einen Fehler feststellst, wende dich an mich .

Siehe auch

Autoren  
Garantie