# ClassAction

Gasmi Salim

| COLLABORATORS | | | |
|---|---|---|---|
| | *TITLE* :<br><br>ClassAction | | |
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | Gasmi Salim | March 26, 2025 | |

| REVISION HISTORY | | | |
|---|---|---|---|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

# Contents

# Chapter 1

# ClassAction

## 1.1 ClassAction 3.1 Guide

```
ClassAction Version 3.1


 Was ist ClassAction
 System Vorraussetzungen

 Installation
 Installation Teil II

 Benutzung von ClassAction

 Konfiguration
 ClassAction Prefs

 Der Arexx Port von CA

 Zukünftige Erweiterungen
 Geschichte (in Englisch)
 Registrierung
 Die deutsche Übersetzung

 Lizenz     Der Autor     Grüße      Bekannte Fehler
```

## 1.2 Was ist ClassAction ?

```
ClassAction ist ein kleines Hilfsmittel, welches das Leben eines
jedes Fesplattenbesitzers vereinfacht.

Wer eine Festplatte besitzt hat immer verschiedenste Dateien:
Programme, Module, Bilder, Sources, Animationen, Sounds...

ClassAction erkennt den Typ der ausgewählten Datei und stellt
eine Liste von auf die Datei ausführbaren Aktionen zur Verfügung.

Zum Beispiel: Wenn ein Bild im GIF Format ausgewählt wird, erkennt
```

ClassAction dieses als GIF Klasse und zeigt eine Liste von zur Verfügung stehenden Aktionen an, wie zum Beispiel 'Anzeigen' und 'Bearbeiten'.

ClassAction ist weitestgehend konfigurierbar. Es können beliebige Klassen und Aktionen hinzugefügt werden.
Aktionen benutzen externe Programme, so daß jeder seine bevorzugten Anzeigeprogramme (usw.) verwenden kann.

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

ClassAction benutzt die xfdmaster.library um gepackte Dateien automatisch zu entpacken. So kann auch eine gepackte Klasse erkannt werden.

ClassAction benötigt sehr wenig Speicher und Rechenzeit.

Bevor Sie sich entschließen ClassAction zu löschen, testen Sie es !!!

Weiter Informationen über ClassAction sind Hier zu finden.

## 1.3 Mehr über ClassAction

ClassAction und ClassActionPrefs sind (C) 1994-96 by Gasmi Salim

Dieses Paket ist Shareware. Fühlen Sie sich frei es zu benutzen !!!
und zu verteilen solange keine Datei des Archivs verändert wird.
Wenn Sie ClassAction regelmäßig benutzen, müssen Sie sich registrieren lassen !!!

PD Vertreiber dürfen das ClassAction Paket in ihre Sammlungen integrieren, solange mir die Integration mitgeteilt wird.

Dies ist Release 3.1 von ClassAction.

Bei Gefallen und Benutzung müssen Sie sich registrieren lassen.

Technische Informationen über ClassAction sind Hier zu finden.

## 1.4 System Vorrausetzungen

Um ClassAction zu benutzen ist folgendes nötig:

- ein Amiga
- Amiga OS 2.0 und höher
- MUI 3.0 und höher
- eine Festplatte
  (ClassAction ist ohne Festplatte nutzlos)

## 1.5 Konfiguration von ClassAction

Die Oberfläache und die Funktionen von ClassAction können durch
ToolTypes konfiguriert werden.

Um einen ToolType Wert zu ändern, muß zuerst das ClassAction Icon
ausgewählt werden, und dann der Menupunkt Informationen aus
dem Piktogramm Menu der Workbench.

Dies ist eine Liste aller ToolTypes:

Achtung: Der Vorgabe Wert für jedes ToolType wird verwendet, wenn
das entsprechende ToolType beim Programmstart nicht gefunden
wird.

==========

DONOTWAIT

Bitte nicht entfernen, dieses ToolType wird benötigt, wenn
ClassAction aus der Schublade WBStartup geladen wird.

==========

DECRUNCH

Wird dieses ToolType aktiviert (DECRUNCH=YES) versucht ClassAction
gepackte Dateien mit Hilfe der xfdmaster.library zu entpacken.

Es wird benötigt, wenn gepackte Klassen erkannt werden sollen.

In diesem Fall, wird die Datei in den Hauptspeicher geladen
und ein Puffer für die entpackte Datei allokiert. Deswegen
benötigt diese Funktion Speicher (ungefähr 2,5 x Dateigröße!).
Sollte ihr Amiga über wenig Speicher verfügen, ist es besser
diese Funktion zu deaktivieren.

Vorgabewert: NO (deaktiviert)

==========

ICONFILE

Dieses ToolType gibt die AppIcon Datei an.
Mit ihr kann ein individuelles AppIcon für ClassAction gewählt
werden. Wird dieses ToolType auf eine .info Datei (icon) gesetzt,
wird ClassAction diese als AppIcon verwenden.

Achtung: Die .info Endung der Icon Datei darf nicht mit angegeben
werden.

Beispiel: ICONFILE=Sys:icons/head
wird das Icon head.info aus dem Verzeichnis Sys:Icons/
als AppIcon verwenden.

Kann ClassAction die angegebene Icon Datei nicht verwenden, so
wird das Vorgabe Icon (ClassAction.info) verwendet.

Vorgabewert: ""  (ClassAction.info wird verwendet)

==========

CLISIZE

Dieses ToolType gibt die Größe des CLI Ausgabe Fensters bei
Verwendung von SystemTags() an.

Der Syntax: DEVICE:OberesX/OberesY/Breite/Höhe/Titel

Somit kann ein anderes Device als das standardmäßige CON:
verwendet werden und die Position und Dimension des Fensters
festgelegt werden.

Achtung: Wird dieses ToolType auf falsche Werte gesetzt, kann
ClassAction das CLI Ausgabefenster nicht öffnen.
Modifizieren Sie dieses ToolType nur, wenn Sie wissen
was Sie machen!
** Nie ** Device Befehle wie AUTO, CLOSE, WAIT einfügen,
dies wird von ClassAction gemacht.
** Nie ** Leerzeichen in den Titel einfügen.

Das ähnliche ToolType OUTPUT aus Version 2.0 von ClassAction
wird nicht mehr benötigt und unterstützt!

Vorgabewert:  CON:0/0/640/100/ClassAction_Output_Window

==========

GTLIKE

Wird dieses ToolType definiert, sieht das MUI Dateiauswahl-
fenster wie das GadTools Auswahlfenster aus. Zum Beispiel:

Verzeichnisse in weiß
Dateien in schwarz
.info Dateien werden dargestellt

==========

CAPREFS

Dieses ToolType enthält den Pfad für das ClassActionPrefs
Programm. Dieser wird benötigt, wenn ClassActionPrefs über
den "Voreinst." Knopf von ClassAction aktiviert wird.

Vorgabewert: "SYS:prefs/ClassActionPrefs"

==========

DRIVE1 bis DRIVE50

Wenn diese ToolTypes auf gültige Pfade gesetz werden, so wird
der Text in der Laufwerksliste von ClassAction angezeigt.
Dies ist nützlich, um schnell in ein Verzeichnis zu gelangen.

Der syntax: DRIVEx=<Schalter Text>,<Pfad>

Beispiel: DRIVE9=Jpeg,dh0:gfx/pictures/jpeg
hier wird der 9. Schalter mit dem Text Jpeg beschrieben
und der Pfad auf dh0:gfx/pictures/jpeg

Beispiel: DRIVE3=dh0:libs
hier wird der Schalter mit demselben Text beschrieben,
wie als Pfad gesetzt wird.

Actung: Die Schaltertexte dürfen nicht zu lange sein, da sie
sonst nicht angezeigt werden.

Vorgabewert: "" (nichts)

## 1.6 ClassAction Prefs

Das Kernstück des Programms ist die Definition von neuen Klassen und
Aktionen. Dies geschieht mit dem ClassActionPrefs Programm.

Das Fenster ist in zwei Seiten aufgeteilt: Klassen & Aktionen.

Nach auswählen einer Klasse werden die zugehörigen Aktionen im
Aktionsteil angezeigt.

Um eine Klasse hinzuzufügen oder zu löschen muß nur auf die entsprechenden
Schalter geklickt werden.

Dasselbe Konzept gilt für die Aktionen...

1. Was ist eine Klasse

2. Erzeugen einer neuen Klasse

3. Definieren einer neuen Aktion

4. Die Lernfunktion

## 1.7 Die Lernfunktion

Die Lernfunktion hat den Zweck beim Definieren neuer Klassen zu
helfen.

Beim Definieren einer neuen Klasse müssen Offsets für diese festgelegt
werden. Es wäre umständlich und langweilig Dateien zu bearbeiten, um
zu erfahren, durch welche Offsets diese neue Klasse bestimmt wird.

Dies ist der Grund für die wunderbare Lernfunktion!

Die Benutzung verläuft nach folgenden Punkten:

1 : Klasse normal definieren, Klassenname und Namensmuster eintragen.

2 : Den 'Lernen' Knopf anklicken.
Die Genauigkeit kann zwischen 75 und 100% eingestellt werden.
Je höher diese eingestellt ist, desto genauer wird die Klassen-
erkennung. Es wird empfohlen immer 100% zu verwenden.

3 : Mit dem bereitgestellten Dateiauswahlfenster soviele Dateien,
die zur neuen Klasse gehören, auswählen wie gewünscht. Je mehr
Dateien angegeben werden, desto besser wird das Ergebnis.
ClassAction versucht daraufhin die Offset Definition zu finden.
(Um mehrere Dateien auszuwählen muß die Shift Taste verwendet werden!)

4 : Nach der Analyse werden die gefundenen Offsets in einem Fenster
angezeigt.

5 : Bei Bedarf können sie von Hand modifiziert werden.

6 : Wenn alles passt und die Offsets verwendet werden sollen
'Übernehmen' anklicken und ansonsten 'Abbruch'.

ACHTUNG: Wenn die Lernfunktion richtig arbeiten soll muß folgendes
beachtet werden:

* Alle ausgewählten Dateien gehören zu ein und derselben Klasse.

* Keine der ausgewählten Dateien ist gepackt.

Das wärs !!!

## 1.8 Was ist eine Klasse

Eine Klasse ist eine Gruppe (Familie) von Dateien. Zum Beispiel
kann man alle C Source Dateien als C Klasse betrachten.

Mit ClassActionPrefs können beliebig viele Klassen erzeugt werden,
solange eine Erkennungsmethode mit festgelegt wird.

Es zwei Arten von Erkennungsmethoden:
- Namensmuster: Die Datei wird anhand ihres Namens einer Klasse
  zugeordnet. (z.B Endung oder Prefix)
- Dateiinhalt: Der Dateiinhalt wird durch Offsets zugeordnet.

Es gibt drei festeingebaute Klassen, die weder gelöscht noch
umbenannt werden können. Sie erscheinen in der Klassenliste mit
weißer Schrift:
- 'Unbekannte Klasse' : Hier werden alle Dateien einsortiert,
  die nicht erkannt wurden.
- 'Grundlegende Aktionen' : Diese Klasse enthält Aktionen die
  in allen anderen Klassen zur Verfügung stehen. Soll zum Beispiel
  eine Aktion 'Kopieren' für alle Klassen definiert werden, so
  kann diese für jede Klasse einzeln definiert werden. (dauert

lange...) Oder aber schnell und einfach in der 'Grundlegende Aktionen' Klasse für alle anderen Klassen auf einmal!
- o 'Verzeichnis' : Enthält Aktionen für Verzeichnisse.

## 1.9 Erzeugen einer neuen Klasse

Um eine neue Klasse zu definieren muß nur auf den "Hinzufügen" Knopf geklickt werden.

Eine Klasse hat 3 Bestandteile:
- einen Namen
- einen Namensmuster
- Offsets

o Der Name ist einfach nur der Klassenname, er kann frei gewählt werden.
ACHTUNG: Ein Klassenname muß eindeutig sein, d.h. es dürfen nicht mehere Klassen mit demselben Namen existieren.

o Das Namensmuster ist ein standard AmigaDos Ausdruck, wie z.B. #?.c , mod.#? , #?.c|#?.h , #?b[a|c] , #?toto? .....

(nähere Informationen zu den Platzhaltern sind im AmigaDos Handbuch zu finden)

ACHTUNG: Der Platzhalter * kann nicht verwendet werden. Stattdessen funktioniert des AmigaDos Äquivalent #?.

Das Namensmuster ist unabhänig von Groß- und Kleinschreibung. Z.B. passt toto.C auf #?.c .

Wenn eine Klasse mit Namensmuster definiert wird, muß sichergestellt sein, daß dieses Muster immer zutrifft.

Beispiel: ist eine Gif Klasse mit dem Namensmuster #?.gif definiert werden alle Dateien mit .gif Endung als Gif Bilder erkannt.

Es muß aber sichergestellt sein, daß alle .gif Dateien Gif Bilder sind, und daß alle Gif Bilder auf .gif enden, sonst funktioniert das Erkennen nicht.

Deswegen sollte das Namensmuster nur unter den folgenden 2 Bedingungen verwendet werden:

- Das Namensmuster ist eine Bijektion auf die Klasse (bestimmt die Klasse eindeutig) z.B ist #?.info ein gut genuges Namensmuster für die Icon Klasse. (mit einer anderen Endung aktzeptiert die Workbench es nicht als Icon)

- Es gibt keine andere Wahl als ein Namensmuster zu verwenden. Wie soll z.B: ein C Source von einer Ascii Datei unterschieden werden, außer über die Endung .c ?

o Ansonsten sollten Offsets verwendet werden.

Ein Offset ist ein Stück in einer Datei, wo nach etwas bestimmten gesucht wird.

Gif Bilder zum Beispiel beginnen immer mit dem Text 'GIF' im Offset 0.

Es gibt Syntaxen für die Offset Definierung:

==========

Syntax #1 : Offset,HexString

- Offset ist eine Dezimalzahl, die den Offset enthält
- HexString ist eine Zeichenkette in hexadezimaler Form, die am Offset gefunden werden soll.

Beispiel: 0,4f4a bedeutet, daß die Datei mit den Bytes $4f und $4a an der Position 0 beginnen muß.
Beispiel: 9,448b3c ab Byte Nummer 9 der Datei wird der Text $44 $8b $3c gefunden.

==========

Syntax #2 : Offset,'String'

- Offset ist eine Dezimalzahl, die den Offset enthält
- String ist eine Ascii Zeichenkette die am Offset gefunden werden soll.

Besipiel: 0,'GIF' Die Datei muß mit der Zeichenkette 'GIF' beginnen

Beispiel: 9,'FuBar' Die Datei hat ab Byte Nummer 9 die Zeichenkette 'FuBar' stehen

==========

Syntax #3 : Offset,"String"

- Offset ist eine Dezimalzahl, die den Offset enthält
- String ist eine Ascii Zeichenkette die am Offset gefunden werden soll.

Die Unterschied zwischen Syntax #2 und Syntax #3 liegt darin, daß in Syntax #3 "" statt '' verwendet werden. Vom Konzept her sind beide identisch, außer, daß Syntax #3 Groß- und Kleinschreibung im Gegensatz zu Syntax #2 nicht berücksichtigt.

Beispiel: Eine AmigaGuide Datei beginnt immer mit: @database
(klein oder groß geschrieben)
Die Erkennung sähe in Syntax #2 z.B. so aus:
0,'@database'. ClassAction würde dann eine mit @DATABASE beginnende Datei nicht als AmigaGuide Datei interpretieren.
Dies funktioniert jedoch, wenn Syntax #3 verwendet wird: 0,"@database"

```
=====================================================================

Es können bis zu 5 Offsets für eine Klassendefinition angeben
werden. Eine Datei wird nur dann in diese Klasse einsortiert, wenn
alle Offsets den richtigen Wert enthalten.

Beispiel: Die Klasse X ist folgendermaßen definiert:
            Offset #1 :  0,4a8b6c
            Offset #2 : 58,14
          Alle Dateien, die mit 4a8b6c beginnen UND an Byte #58
          $14 stehen haben werden in diese Klasse einsortiert.

Um das nächste von meheren Offsets zu aktivieren reicht es das
Cycle Gagdet 'Offset #' anzuwählen.

Bemerkung : Die Klasse 'Ascii Datei'

Ed gibt einen festeingebauten Offset Befehl namens ASCII[]

Wird dieser Befehl in Offset #1 gesetzt, dient dies zur Erkennung
von Ascii Dateien.
Aber ClassAction wird diese Klasse als allerletzte testen.
Deswegen werden AmigaGuide Dateien (Die Ascii Dateien sind)
nicht als Ascii Dateien erkannt, solange eine AmigaGuide
Klasse definiert ist.

Normalerweise sollte dieser Befehl nicht verwendet werden, da
in der mitgelieferten Standard Prefs Datei eine Ascii Klasse
über diesen Befehl definiert ist.
```

## 1.10 Definieren einer neuen Aktion

```
Nahc erzeugen einer Klasse sollten für diese Aktionen definiert
werden. Jede Klasse kann beliebig viele Aktionen erhalten.

Dazu muß nur der 'Hinzufügen' Knopf angeklickt werden.

Eine Aktion hat 6 Bestandteile:

    - einen Namen
    - einen Ausführungsmodus (mode)
    - die Stack Größe (nur wenn der Ausführungsmodus CLI ist)
    - eine Verzögerung (ebenfalls nur bei CLI)
    - ein Programm
    - die ReScan Dir Option

o   Name stellt den Namen der Aktion dar.

o   Ausführungsmodus kann sein :  Cli ,  WB ,  No Cli
    oder  ARexx .

o   Programm ist eine Befehlszeile in AmigaDos Format und
    kann Parameter enthalten. Für zu startende Programme sollte immer
    der komplette Pfad verwendet werden.
```

Beispiel: C:Copy anstatt von nur Copy

Die Programm Zeile kann Argumente und Befehle , die
fest in ClassAction eingebaut sind enthalten.

- Die ReScan Dir Option sollte gesetzt werden, wenn die Aktion das aktuelle VErzeichnis neu einlesen soll. Dies ist sinnvoll bei Aktionen die das Verzeichnis verändern, wie zum Beispiel 'Löschen' oder 'Umbenennen'.

Knöpfe

- Mit den Pfeilen hoch und runter können die Aktionen sortiert werden.
- Der 'Laden' Knopf liefert ein Dateiauswahlfenster, in dem ein Programm für die Programmzeile ausgewählt werden kann.
- Der Befehle Knopf zeigt eine Liste mit allen möglichen Argumenten und Befehlen für die Programmzeile.
- Eine bereits existierende Aktion kann folgendermaßen in die aktuelle kopiert werden:
  - Anklicken des Knopfes rechts vom Aktionsnamen.
  - die Klasse der gewünschten Aktion in der Verzeichnis Baum Liste auswählen (durch Doppelklick darauf)
  - und dann die gewünschte Aktion doppelklicken

## 1.11 CLI Modus

**** CLI Modus ****

Wenn 'CLI' ausgewählt wird, dann wird beim auswählen der Aktion diese aus dem CLI gestartet. Dabei wird der Stack Wert auf den angegebenen Wert gesetzt. (Vorgabe ist 4096)

Dieser Modus wird jedoch nur bei Bedarf (wenn irgendetwas ausgegeben wird) eine CLI Fenster öffnen.

Für das CLI Fenster kann eine Verzögerung eingestellt werden:

Bei negativer Verzögerung (z.B. Verzögerung = -1) wartet das CLI Fenster solange, bis es von Hand geschlossen wird.

Bei Verzögerung 0 schließt sich das Fenster von selber sobald die Aktion beendet ist.

Bei positiven Verzögerungswerten wartet das CLI Fenster mit dem Schließen entsprechend viele Sekunden. Das Fenster kann jedoch schon früher von Hand geschlossen werden.

Die Dimensionen des CLI Fensters können mit dem ToolType CLISIZE bestimmt werden.
(das alte ToolType OUTPUT wird nicht mehr unterstützt).

## 1.12 WB Modus

```
*** WB Modus ***

Ist dieser Modus ausgewählt, wird kein CLI geöffnet. ClassAction
simuliert dann einen Start von der WorkBench aus. Das Programm
wird dann mit Werten aus seinem Icon gestartet.

ACHTUNG: Dieser Modus ist nur brauchbar mit Dateien, die Icons
         haben.
```

## 1.13 NO CLI Modus

```
*** NO CLI Modus ***

Hierbei wird das Programm vom CLI aus gestartet. Jedoch wird
in keinem Fall ein Fenster geöffnet.
```

## 1.14 ARexx Modus

```
*** AREXX Modus ***

Dieser Modus startet rx mit dem gegebenen Programm. Das gegebene
Programm MUSS eine ARexx Datei sein. Zusätzlich muß RexxMaster
aktiviert sein, und Rx im Verzeichnis Sys:rexxc liegen, damit
das ARexx Skript ausgeführt werden kann.
```

## 1.15 Argumente

```
Momentan existieren 8 Argumente:

Die ersten 4 Argumente sind kleingeschrieben : [f] [s] [b] [x]
und sie liefern das Ergebnis in Anführungszeichen zurück.

[f] : komplette Pfad der ausgewählten Datei in Anführungszeichen
[s] : dasselbe, aber ohne eventuelle Dateiendung
[b] : Dateiname der ausgewählten Datei in Anführungszeichen
[x] : genauso, nur ohne eventuelle Endung

Die selben Argumente großgeschrieben bewirken das Gleiche, nur
ohne Anführungszeichen.

[F] : komplette Pfad der ausgewählten Datei
[S] : dasselbe, aber ohne eventuelle Dateiendung
[B] : Dateiname der ausgewählten Datei
[X] : genauso, nur ohne eventuelle Endung

Beispiel: Die ausgewählte Datei ist Ram:env/sys.prefs
```

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

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

Beispiel: Die ausgwählte Datei ist Ram:main.c

* Die Programmzeile
            c:copy [f] [F].bak
  wird ersetzt durch:
            c:copy "Ram:main.c" Ram:main.c.bak

* Die Programmzeile
            c:copy [f] [S].bak
  wird erzetzt durch:
            c:copy "Ram:main.c" Ram:main.bak
```

## 1.16 Befehle

```
Aktuell sind folgende 5 Befehle möglich:

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 den Benutzer nach Fortsetzung.

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

REQs Befehl:

    Die REQs Befehle öffnen ein ReqTools Fenster mit dem Titel [text].
    Dies ist für interaktive Befehlszeilen nützlich.

    Beispiel:
        bin:lha x [f] to REQD[Zielverzeichnis:]

        Dadurch wird ein Auswahlfenster geöffnet, welches dem Benutzer
        erlaubt, daß Zielverzeichnis zum Entpacken anzugeben.

    REQF[] macht dasselbe, nur wird hier nach einer Datei gefragt wird.

    Beispiel:
        c:dir [f] > REQF[Zeiledatei:]


    REQV[] fragt den Benutzer nach einem Laufwerk.
```

```
    REQT[] fragt nach einem Text, der z.B. als Argument dienen kann:
        c:cpu REQT[Argumente für CPU:]


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

SURE Befehl:

    SURE[text] wird ein Fenster mit dem Text [text] und den Schaltern
    Ja und Nein öffnen.
    Wird Nein ausgewählt, wird die Ausführung der Programmzeile
    abgebrochen. Bei Ja wird der Rest der Programmzeile ausgeführt.

    Beispiel:
        SURE[Wirklich die Datei löschen?]C:delete [f]

        In diesem Beispiel wird der Benutzer gefragt, ob er die Datei
        wirklich löschen will. Bei Nein wird nichts gemacht, bei Ja
        wird C:delete [f] ausgeführt.

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

Es können beliebige Argumente und Befehle kombiniert werden.

Beispiel:
    SURE[Datei wirklich umbennen?]c:rename [f] REQF[Neuer Dateiname:]
```

## 1.17 Zukünftige Erweiterungen:

```
Was Ich in der nächsten Version des Programms noch hinzufügen will:

o   Konzept von Familien von Klassen:

    Zum Beispiel: Bilder Familie, die GIF, IFF, TARGA, JPEG...
    Klassen enthält. Ein Familienfilter im Verzeichnis Auswahlfenster.
    So könnten Bilder,... ausgefiltert werden.

o   Und natürlich alles wonach ich gefragt werde :)
```

## 1.18 Über

```
Danksagungen gehen an folgende Leute:

o   Mireille (für ihre Geduld...)

o   Philippe Thomas (für Vorschläge, Hilfe, Beta testing,
    Die fränzösische guide Datei und UTT, verwendet für die
    Installation)

o   Bruno Durremberger (für Beta testing)

o   Jean Michel Dessolas  (für Beta testing auf A4000/40)
```

- Obvious Implementations Corp (für Dice C Pro)
- Nico Francois für die ReqTools library
- Georg Hörmann für die *GROSSARTIGE* xfdmaster library
- Stefan Stuntz für das wunderbare Magic User Interface .
- Alle Benutzer die mir Fehlerberichte und Vorschläge geschickt haben.
- Alle registrierten Benutzer.

## 1.19 Der Author

Meine Anschrift:

Gasmi Salim
6, rue des Hirondelles

67380 Lingolsheim
France

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

E-Mail: salim@sdv.fr

## 1.20 Benutzung von ClassAction

Die Benutzung von ClassAction ist wirklich einfach.

Es muß nur eine Datei in dem zur Verfügung gestellten Dateiauswahlfenster ausgwählt werden. Mit dem Schalter 'Mutterv.' gelangt man ins nächst höhere Verzeichnis.

Danach wird in der rechten Liste die entsprechende Klasse und die dazugehörigen Aktionen angezeigt.
Es muß nur noch die gewünschte Aktion ausgwählt werden.

Beim Doppelklick auf eine Datei wird die oberste Aktion direkt ausgeführt.

Das ClassAction Fenster ist ein AppWindow. Es können Dateien direkt von der Workbench in das Hauptfenster gezogen werden.

Mit dem 'Voreinstellungen' Knopf kann direkt das Voreinstellungsprogramm ClassActionPrefs aufgerufen werden.

'Ende' beendet das Programm.

Um das Fenster in ein AppIcon zu verwandeln muß es nur geschlossen
werden. Zum Wiederherstellen des Fensters reicht dann ein Doppel-
klick auf das AppIcon.

Ist ClassAction im AppIcon Zustand können ebenfalls Programme direkt
auf das AppIcon gezogen werden. Ist in der entsprechenden Klasse
nur eine Aktion definiert wird diese direkt gestartet, ansonsten
wird ein Auswahlfenster für die verfügbaren Aktionen zur Verfügung
gestellt.

Die Benutzung des Dateimanagers ist ebenfalls einfach. Die linke
Verzeichnisliste ist immer die Quellliste und die rechte Liste in der
'Bearbeiten Gruppe' das Zielverzeichnis.

Es existieren einige Aktionen für den Dateimanager wie 'Kopieren',
'Umbenennen',... , die einfach zu handhaben sind.

Der Knopf '<-->' vertauscht die Pfade in des Qell- und Zielverzeichnisses.

.info Dateien werden vom Dateimanager nicht automatisch mit bearbeitet
(kopiert, umbenannt...)

Der Dateimanager sollte als "Bonus" und nicht als komplettes Programm
aufgefasst werden!

## 1.21 Technische Informationen

ClassAction ist zu 100 % mit DICE C 3.0 geschrieben.

Verschiedenes:

Die Prefs Datei ist eine ASCII Datei namens : ENVARC:ClassAction/ClassAction. ↩
prefs
Die Grundlegenden Aktionen werden unter ENVARC:ClassAction/ClassAction_Gen. ↩
prefs gesichert.
Die Verzeichnis Aktionen werden unter ENVARC:ClassAction/ClassAction_Dir.prefs ↩
gesichert.

ClassAction erzeugt eine ausführbare Datei namens ClassAction_RunTask in T:
Dieses Programm wird zum ausführen von WB Programmen benutzt.

Library Informationen :

ROM libraries benötigt:

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

```
DISK libraries benötigt:

    rexxsyslib.library      V39+
    commodities.library     V37+
    asl.library             V39+
    icon.library            V37+
    reqtools.library        V38+
    MUI libs                V3.0+


DISK libraries benötigt falls vorhanden:

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

Erkennung einer Klasse:

    1. Test auf passendes Namensmuster
    2. Test auf passende Offsets
    3. Entpacken mit Hilfe der xfdmaster.library
    4. Test auf passende Offsets in der entpackеten Datei
    5. Test, ob es sich um eine Ascii Datei handelt (wenn der Ascii
       Offset Befehl existiert)

    Wenn alles nichts bringt: Einstufung als 'Unbekannte Klasse'
```

## 1.22 Installation

```
Um dieses Programpaket zu installieren :


Einfach das mitgelieferte Installer Skript durch Doppelklick
auf das Instalations Icon starten.
```

## 1.23 Registrierung

```
Warum soll man sich registrieren lassen?

Als erstes, um den besten Computer, der je gebaut wurde
zu unterstützen. Den die Zukunft des Amiga hängt von der
Verfügbarkeit von Software ab. Durch unterstützen eines
Programmierers für seine Arbeit unterstützt man den Amiga
und seine Zukunft!

Desweiteren habe ich viel freie Zeit auf diesen Versuch
Shareware zu programmieren verbracht. Also warum sollten Sie
nicht bei Benutzung die Registrationgebühr bezahlen?
Dies wird mich dazu ermuntern weitere Shareware zu programmieren.
```

Die ShareWare Version ist 100% funktionstüchtig, es wurde nichts
deaktiviert, einzig und allein ein "Nervrequester" wurde eingebaut.

Die Registrierungsgebühr beträgt:

        10 US$ oder 20 DM oder 50 FF.

Senden Sie ihre Gebühr an meine  Adresse ,
und Sie werden die neueste registrierte Version von ClassAction
erhalten.

Vergessen Sie nicht ihre VOLLE Adresse (inklusive Land) anzugeben.

        Danke im voraus für Ihre Unterstützung!

                Ihr,

                        Salim

## 1.24 Legal words

Copyright

ClassAction und ClassActionPrefs sind Copyright © 1994-1996
by Gasmi Salim.

ClassAction ist ein Shareware Programm. Das Paket darf in keiner
Weise verändert und darf ohne die vorherige schriftliche Genehmigung
des Autors nicht für kommerzielle Zwecke verwendet werden. Die
Copyright Nachricht muß erhalten bleiben.

Garantie

Es wird keine Verantwortung für irgendeinen Schaden der von der
Benutzung dieses Programmes herrühren könnte übernommen. Jede
Benutzung ist auf eigenes Risiko. Die Software wird vertrieben
als "as is" ohne irgendwelche Garantien für Funktion
und Genauigkeit der Software und Dokumentation. Die Dokumentation
scheint korrekt zu sein, der Auto behält sich jedoch das Recht vor
die Software und/oder Dokumentation ohne Bekanntgabe zu ändern.

## 1.25 History

        ClassAction History V 3.1 (c) Salim Gasmi

09/04/96 : V3.1

- ClassAction hat jetzt einen kleinen eingebauten Dateimanager.

- Einige Fehler wurden behoben.

12/03/96 : V3.0 (Major Update)

- ClassAction und ClassActionPrefs verwenden jetzt MUI 3.0+

- ClassAction unterstützt bis zu 50 Laufwerksknöpfe

- ClassAction erkennt Verzeichnisse

- ClassAction beerabeitet Mehrfachauswahl

- ClassAction hat ein AppWindow

- Die Genauigkeit der Lernfunktion kann eingestellt werden

- Tooltypes CAPREFS,GTLIKE hinzugefügt.

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

25/09/95 : V2.8  ***** ab hier nicht mehr übersetzt *****

- ClassAction have now a FULL commodtity support
  and a HotKey to show/hide ClassAction.

- Tooltype APPSTART can be set to HIDE (APPSTART=HIDE)
  if you want that ClassAction start hidden.

- Tooltype CX_HOTKEY added

- Tooltype PUBSCREEN added to allow using Public Screens

- The Internal File Selector of ClassAction was not freeing all
  the memory allocated, fixed now .

11/09/95 : V2.75

- The REQD,REQF,REQV commands were not really incompatibles
  with the appicon mode, but incompatibles with some programs
  such MagicMenu (Bad luck I use MagicMenu...)

  A lot of users complained about the automatic swith into
  a REQT command even if they don’t use an incompatible program.

  I have added a tooltype (REQBUG) to let the user choose to
  auto-switch into REQT or not.

03/09/95 : V2.7

- Added commands [b],[x],[B],[X],[F],[S]

- Added the Action arexx command.

- removed some bugs in rendering routines

- removed a bug with multiple icons throwed on the AppIcon.

- The REQD,REQF,REQV commands are now swapped into REQT
  when using them from the AppIcon or Arexx , they are only
  compatible with the Window mode.

28/08/95 : V2.6

- ClassAction window is now resizable.

- ClassAction and ClassActionPrefs are now using ReqTools.library.

- Added REQV command to request a volume.

- Added REQT command to request a text.

- Added WINX and WINY tooltypes.

- Learn requester has now an ALL button.

- Selected File is now in all the REQs requesters.

- ClassAction use now a Key file for the registerd versions,
  this key file is placed in S: .

17/07/95 : V2.5

- ClassAction and ClassActionPrefs are now localised.
  and a French catalog is provided with the archive.

- Learn function added to ClassActionPrefs.

- To select the first action of a file, you must now Double Click
  on it instead of reselecting the file.

- Generic Actions are now Synchro and rescans the current directory.

- AUTOSELECT ToolType is no more used.

- REQs requesters opens now in current directory.

- minor improvements made.

12/06/95 : V2.1

- Added the 'Generic Actions' Built-in Class.

- Added SURE[] exec command.

- Added ASCII[] Offset Command to recognize ASCII files.

- Added Arexx commands: AppIconify, Show, Status, GetClass.

- Changed running tasks system, now I use Systemtags().
  We do not need tmp files anymore.

- you can now define a delay for CLI run mode.

- Added 'string' and "string" for offsets definition.

- ToolType OUTPUT is obsolete now and not used anymore,
  we use the new ToolType CLISIZE in replacement.

- Actions requester is now well sized, appear below the
  mouse pointer and uses to frontmost public screen.

- Swapped the buttons 'Use' and 'Save" and added 'Cancel'
  in ClassActionPrefs to follow the Amiga prefs look,
  moved the button 'about' in top right corner as '?'.

- ClassActionPrefs Cycle gadgets routines weren't
  100 % system friendly and some patches like Cycle2Menu
  makes bugs with ClassActionPrefs; It's fixed now.

- Improved the recognizer code and the Info routine:
  they are up to 400% faster.

- Listview hilight color error removed.

- ClassAction does not anymore lock the Workbench
  screen when AppIconified.

- A nasty bug found and removed in ClassActionPrefs.

- Right Mouse button shows Assigns only if mouse
  is in the requester and Right Mouse again brings
  back to the Directory.

- ClassAction remember now the window position.

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

- ClassAction has now an AppIcon.

- ClassAction is now a commodity.

- ClassAction has now an Arexx port.

- ClassAction use now the default WB Font.

- Exec mode 'Arexx' added.

- Different color for Directories/Files.

- Up/Down gadgets added to ClassActionPrefs.

- 'Use' gadget added to ClassActionPrefs.

- Classes are now sorted into ClassActionPrefs Listview.

- APPSTART, ICONNAME, ICONX, ICONY, CX_PRIORITY,
  WBFONT, OUTPUT, ICONFILE ToolTypes added.

- When the window is iconnified it have now the right height
  regarding the screen default font.

- New Save Format (CASF20).

- Suffix/Prefix Button removed. Replaced by MatchName Gadget
  who accept any Wildcard.

- Config file moved into ENVARC:

- an installer is now provided with the archive.

- some Code optimization done.

05/05/95 : V1.43

- Cleaned up the requester code, now 5% faster.

- ClassAction now look for his name using WBstartup structure
  and then can be renamed .

02/05/95 : V1.42

- 'No Cli' Exec mode added to ClassActionPrefs.

- "" are always added to filenames even if not needed
  it's easier for AREXX scripts.

06/04/95 : V 1.4

- ToolType HEIGHT added.

- Some code optimization added.

22/02/95 : V 1.31

- If you click twice on the same file the first action
  will be lanched (it's faster than selected the first action
  by hand).

- this version is now ShareWare and you must register to get
  the registered version.

15/01/95 : V 1.3

- Added REQD[] and REQF[] interactive commands.

- Copy gadget added to ClassActionPrefs.

- minor improvments made.

- Beta testers reported this version is really stable.

25/11/94 : V 1.22

- First Public Release.

- Button 'Info' Added.

- ClassAction was Locking() the directories it read without UnLocking() them .... *FIXED*

- Code Optimization.

- Minor other Bugs removed.

08/11/94 : V 1.21

- Program was crashing with empty floppy units .. *FIXED*

- Volume/Name Bug Fixed

- <..> Item removed when root of a volume.

07/11/94 : V 1.2

- New interface, I have included my own fast file requester.

- STARTDIR & DRIVE1 to DRIVE11 ToolTypes added.

01/11/94 : V 1.1

- Now using xfdmaster library to recognize and decrunch files.

- Configuration with ToolTypes added (DECRUNCH,AUTOSELECT).

- 'Unknown Class' is now a built in class with unlimited actions.

- New save format (CASF11).

16/10/94 : V 1.0

- New Save Format (CASF10).

- Window has now a zoom gadget.

- a lot of classes definitions added.

```
10/10/94 : Beta Version

    - tmp file bug removed.

    - Offset increment error removed.

    - using asl library for the file requester.

01/10/94 : Alpha Version
```

## 1.26 Die Arexx Befehle

```
    ClassAction hat einen Arexx Port mit dem Namen: ClassAction.01

    Hier eine Liste aller ARexx Befehle:

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

    Quit

        Beendet ClassAction...

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

    Use

        Zwingt ClassAction die Prefs Datei erneut zu laden.

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

    Ver

        Gibt die Version von ClassAction zurück.

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

    Status

        Gibt den Aktuellen Status von ClassAction zurück.

        Rückgabewert 0 : ClassAction ist im AppIcon Zustand.
        Rückgabewert 1 : ClassAction hat das normales Fenster offen.

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

    AppIconify

        Zwingt ClassAction in den AppIcon Zustand überzugehen.

        Ist ClassAction bereits in diesem Zustand, bewirkt dieser
        Befehl nichts.
```

```
=====================================================================

Show

    Zwingt ClassAction sein Hauptfenster zu zeigen.

    Ist ClassAction bereits in diesem Zustand, bewirkt dieser
    Befehl nichts.

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

Load <filename>

    ClassAction versucht die Datei <filename> zu laden und öffnet
    das Aktionenfenster, damit der Benutzer eine Aktion auswählen
    kann.
    Existiert die Datei nicht, bewirkt dieser Befehl nichts.

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

GetClass <filename>

    ClassAction versucht die Datei <filename> zu laden und gibt
    den Klassennamen der geladenen Datei zurück.

    Existiert die Datei nicht, bewirkt dieser Befehl nichts.

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

Action <filename> <Action Pattern>


    ClassAction führt die erste zu Action Pattern passende Aktion
    für die Datei <filename> aus.

    Besipiel: Action ram:toto.lha extr

              führt die erste Aktion, deren Name den Text 'extr'
              enthält auf der Datei ram:toto.lha aus.

    Das 'Action Pattern' hat den Zweck, das nicht der Exakte Name
    der auszuführenden Aktion angegeben werden muß, es reicht
    ein Teil.
```

## 1.27 Installation Teil II

```
Nach der Installation sollte eine funktionstüchtiges ClassAction
vorliegen mit vielen Klassendefinitionen, aber mit sehr wenigen
Aktionen dazu. (Viele Klassen haben gar keine fertig definierten
Aktionen)

Es liegt am Benutzer Aktionen, die zu seiner Systemkonfiguration
und zu den vorhandenen Programmen passen zu definieren.
```

Um eine an die persönlichen Bedürfnisse angepasste Konfiguration
zu bekommen muß nur ClassActionPrefs gestartet werden und alles
konfiguriert werden...

Information über die Konfiguration von Klassen und Aktionen enthält
diese Guide Datei.

Nach einer Konfiguration der Klassen und Aktionen sollte noch eine
Feinabstimmung an den ToolTypes vorgenommen werden.

Es kann lange dauern bis die Konfiguration richtig passt, aber
dann ist ClassAction wirklich großartig !!!

## 1.28 Bekannte Fehler

ClassActionPrefs darf nicht gleichzeitig öfters gestartet werden,
sonst gibts Prolbleme... (GURU)

## 1.29 Die deutsche Übersetzung

Die deutsche Übersetzung

Diese Übersetzung ist auf die mitgelieferte .catalog Datei abgetimmt.
Sie hält sich weitestgehend an das Original, hat aber einige nicht
näher gekennzeichnete Anmerkungen und Erweiterungen, die im Original
nicht enthalten sind.
Rechtsschreibfehler bitte ich zu entschuldigen.
Für Vorschläge und Verbesserungen hier meine email:

hans@informatik.tu-muenchen.de

bzw. meine komplette Anschrift:

Martin Hans
Römerstraße 17
85232 Bergkirchen

Viel Spaß mit ClassAction

Martin