

ClassAction

Gasmi Salim

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ClassAction	1
1.1	ClassAction 2.8 Guide	1
1.2	Vad är ClassAction ?	2
1.3	mer om ClassAction	3
1.4	Systemkonfiguration	3
1.5	Konfiguration	3
1.6	ClassAction Prefs	8
1.7	Att använda Lär-funktionen	8
1.8	Vad är en Klass ?	9
1.9	Skapa en ny Klass	10
1.10	Skapa en ny Handling	12
1.11	CLI mod	13
1.12	WB mod	14
1.13	Ej CLI mod	14
1.14	ARexx mod	14
1.15	Argument	14
1.16	Kommandon	15
1.17	Framtida förbättringar	17
1.18	Om	17
1.19	Författaren	18
1.20	Using ClassAction	18
1.21	Teknisk information	19
1.22	Installation	20
1.23	Registrering	20
1.24	Lagliga ting	21
1.25	Historik	21
1.26	Arexx kommandon	26
1.27	Vad du skall göra efter installationen	28
1.28	Meddelande till Philippe THOMAS	29
1.29	Kända buggar	29

Chapter 1

ClassAction

1.1 ClassAction 2.8 Guide

Version 2.8 ClassAction

Vad är ClassAction
Läs detta först

Systemkonfiguration
Vad du behöver

Installation
Hur ClassAction installeras

Efter installationen
Installation, del II

Använda ClassAction
Hur detta verktyg används

Konfiguration
Ändra 'ToolTypes'

ClassAction Prefs
Konfigurera Klasser & Handlingar

Arexx kommandon
AREXX-porten & kommandon

Framtida förbättr.
Vad som skall göras i framtiden

Historik & särdrag

Från början...

Registrering

Hur att registerara sig och varför

Licens

Författaren

Hälsningar

Kända buggar

1.2 Vad är ClassAction ?

ClassAction är ett litet verktyg som förenklar livet alla hårddiskenvändare.

När du har en hårddisk, har du alltid en samling filer:
program, moduler, bilder, källkoder, animeringar, ljud...

ClassAction tar reda på filtyp och visar
en lista över handlingar som kan göras med filen.

Till exempel: När du väljer en GIF-bild komemr den att kännas igen
av ClassAction som en fil av GIF-Klass och en lista över handlingar
kommer att visas som 'Visa' och/eller 'Redigera'.

ClassAction mycket konfigurerbar, du kan lägga till egna
klasser och handlingar.
Handlingar använder utomstående program, detta innebär att du kan
använda det GIF-visarprogram du föredrar bäst (eller vad som helst).

ClassAction har en Applkon, en Arexx-port, är lokaliserat och är
en commodity.

ClassAction använder xfdmaster.library för att auto-dekomprimera
komprimerade filer.
Med denan funktion kan även komprimerad filer klassas.

ClassAction använder väldigt lite minne och CPU-tid.

Om du tar dig tid att konfigurera ClassAction, kan du göra
allting med det !!! Det är ett enkelt sätt att hantere filer.

Innan du raderar det, pröva det !!!!!!!

Om du vill veta mer om ClassAction använd
Mer

1.3 mer om ClassAction

ClassAction och ClassActionPrefs är (C) 1994-95 av Gasmi Salim

Detta arkiv är ShareWare. Känn dig fri att använda det !!!
Och sprid det så länge du inte modifierar någon fil i arkivet.
MEN om du använder det regelbundet måste du registrera dig !!!

PD-förläggare är tillåtna att inkludera ClassAction-arkivet
i deras samling så länge de låter mig veta att de har
inkluderat det.

Detta är utgåva 2.8 av ClassAction.
Om du gillar och använder det MÅSTE du
registrera
dig.

Jag hoppas att du tycker att detta verktyg är användbart.
(Åtminstone jag tycker att det är användbart :))

För teknisk information om ClassAction, använd
Teknisk info

1.4 Systemkonfiguration

För att använda ClassAction behöver du följande saker :

- o Amiga OS 3.0 eller bättre
(Kanske kommer en OS 2.0 version att släppas)
- o En hårddisk
(ClassAction är oanvändbart utan hårddisk)

Det var allt !!!

1.5 Konfiguration

ClassAction gränssnitt och inställningar är konfigurerbara
via verktygstyper (sk ToolTypes).

För att ändra ett ToolType-värde, välj ClassAction-ikonen och
välj 'Information' i 'Ikon'-menyn i WorkBench.

Följande är en lista över alla Verktygstyper:

Obs : Standardvärdet för varje beskriven verktygstyp används om dess Verktygstyp ej kan hittas i ikonen.

DONOTWAIT

Ta INTE bort denna verktygstyp ! Den behövs om ClassAction startas från WBStartUp.

CX_PRIORITY

Detta är Commodity prioriteten för ClassAction.

Standardvärde : 0

CX_HOTKEY

Denna verktygstyp anger vilken sk "Hotkey" som skall öppna/gömma ClassAction.

Om du använder denna Hotkey när ClassAction är gömd eller AppIkonifierad kommer fönstret att öppnas. Om du använder den när fönstret är öppnat, kommer ClassAction att gömmas.

Standardvärde : LALT C (VänsterAlt + Skift + c)

REQBUG

När du använder vissa program såsom MagicMeny (eller andra) kan dessa inkräkta med REQ-kommandot när ClassAction är i AppIkon mod. Systemet krashar oftast...

Om du sätter denna verktygstyp till YES (ex: REQBUG=YES) kommer ClassAction att översätta REQV, REQF och REQD-kommandon till ett REQT-kommando om Handlingen startas från AppIkonen.

Standardvärde : NO

APPSTART

Sätt denna verktygstyp till YES, NO eller HIDE.

NO : ClassAction startas med fönstret öppet.

YES : ClassAction startar som en AppIkon.

HIDE : ClassAction startar gömt.

Standardvärde : NO

=====
STARTDIR

Denna verktygstyp talar om vilket bibliotek som du vill se första gången du kör ClassAction.

Standardvärde : Nuvarande bibliotek

=====
WBFONT

Sätt denna verktygstyp till YES (ex: WBFONT=YES) om du vill att ClassAction skall använda nuvarande Workbench-typsnitt. Om du sätter det till YES, måste typsnittsbredden vara mindre än 15, om den inte är det kommer ClassAction att använda topaz 8-typsnittet.

Om denna verktygstyp sätts till nej, används topaz 8.

Standardvärde : YES

=====
DECRUNCH

Om du sätter denna verktygstyp till YES (ex: DECRUNCH=YES), försöker ClassAction att dekomprimera komprimerade filer med hjälp av xfdmaster.library.

Detta är nödvändigt om du vill kunna använda komprimerade filer.

Om du gör detta, kommer (givetvis) filen att laddas in i minnet och en buffer för dekomprimerade data kommer att allokeras. Därför går det åt minne (ungefär 2.5 ggr filens storlek). Så om du inte har massvis med minne bör du stänga av denna funktion.

Standardvärde : NO

=====
HEIGHT

Denna verktygstyp bestämmer höjden på ClassAction's fönster.

Minimihöjden är beroende av ditt WB-typsnitt. Om du anger en för liten höjd, kommer ClassAction att välja en så liten höjd som är möjligt.

Standardvärde : 0 (Tvångssätter minsta möjliga höjd)

=====

WINX and WINY

Dessa verktygstyper är fönstrets koordinater.

Standardvärden : 10

=====

ICONX and ICONY

Dessa verktygstyper anger AppIkonens koordinater.

Notera att detta bara är en förfrågan till Workbench. Om de angivna koordinaterna redan används av en annan ikon kommer Workbench att ignorera dessa och placera AppIkonen så nära den angivna positionen som möjligt.

Standardvärden : Ingen fast ikonposition.

=====

PUBSCREEN

Om du sätter denna verktygstyp till YES (PUBSCREEN=YES), kommer ClassAction's fönster att öppnas på den främsta, publika skärmen.

Standardvärde : NO

=====

ICONNAME

Denna verktygstyp anger AppIkonens text.

Tex: om du anger ICONNAME=Släpp, kommer texten under AppIkonen vara 'Släpp'.

Texten kan vara tom (ICONNAME=) så att ingen text anges under AppIkonen.

Standardvärde : ClassAction

=====

ICONFILE

Denna verktygstyp anger AppIkonfilen.

Därmed kan du välja din egen AppIkon för ClassAction.

Om du anger en ikonfil mha denna verktygstyp kommer ClassAction att använda den som AppIkonbild.

OBS: Ange ej .info-filtypen i filnamnet.

Exempel: ICONFILE=Sys:icons/head
resulterar i att ikonen head.info från biblioteket
Sys:Icons/ kommer att användas till AppIkonen.

Om ClassAction misslyckas med att läsa din ikonfil används

den ursprungliga standardikonen (ClassAction.info).

Standardvärde : "" (Använd standardikonen)

=====

CLISIZE

Detta är storleken på CLI-fönstret när SystemTags() används.

Syntaxen är : DEVICE:ToppX/ToppY/Bredd/Höjd/Titel

Därmed kan du välja en annan enhet än CON: för CLI och definiera en ny fönsterposition och dimension.

VARNING:

Om du ställer in ett konstigt värde för denna verktygstyp, kommer ClassAction ej att starta något CLI. Modifiera INTE denna inställning om du inte vet vad du gör.

Lägg ** ALDRIG ** till enhetskommandon såsom AUTO, CLOAS eller WAIT till denna sträng, ClassAction sköter om detta åt dig.

Använd ** ALDRIG ** mellanslag i titelsträngen.

I version 2.0 fanns en liknande verktygstyp, OUTPUT, som ej används längre.

Standardvärde : CON:0/0/640/100/ClassAction_Output_Window

=====

DRIVE1 tom DRIVE11

Sätt dessa verktygstyper till en giltig sökväg så visas de som knappar i fönstret.

Dessa används för att snabbt hoppa till ett visst bibliotek.

Syntaxen är : DRIVEx<Knapptext>,<Sökväg>

Tex: DRIVE9=Jpeg,dh0:gfx/pictures/jpeg

På detta sätt kommer knapptext nummer 9 bli 'Jpeg' men sökvägen blir satt till dh0:gfx/pictures/jpeg.

Du kan också ange bara knapptexten utan sökväg,
tex DRIVE3=dh09:libs

I detta fall blir knapptexten densamma som sökvägen.
Använd dock inte för långa strängar, de kommer inte att visas i små knapparna.

Standardvärde : "" (ingen)

1.6 ClassAction Prefs

Att definiera nya klasser och handlingar är hjärtat i ↔
programmet.

För att göra detta, måste du använda ClassActionPrefs-programmet.

Att använda ClassActionPrefs är enkelt... så kör igång !!

Fönstret är delat i två delar : Klasser & Handlingar.

Välj först en klass, dess handlingar kommer att visas i delen
för handlingar.

För att lägga till en ny klass eller ta bort en klass, klicka
på respektive knapp.

Samma koncept gäller för handlingarna...

1. Vad är en Klass
2. Skapa en ny Klass
3. Definiera en ny handling
4. Lär-funktionen

1.7 Att använda Lär-funktionen

Lär-funktionen finns för att hjälpa dig att skapa nya klasser.

När du skapar en ny klass behöver du kanske definiera offseter för
den. Det kan ta lång tid och vara väldigt tråkigt att redigera
filer för att gissa dessa offseter som skall beskriva klassen.

Av den anledningen har den fantastiska funktionen lär skapats !

För att använda den, följ denna procedur :

- 1 : Skapa klassen som vanligt, filnamn och klassnamn.
- 2 : Klicka på 'lär'-knappen.
- 3 : Välj så många filer du vill mha fildialogen så länge
som de tillhör den klass som du skapar.
Ju fler filer du väljer, desto bättre resultat.
ClassAction försöker reda ut vilka ofsseter som behövs
för just denna klass.
(För att välja flera filer, använd skifttangenten.)

- 4 : Ett fönster med de upphittade offseterna visas efter analysen.
- 5 : Redigera dem manuellt om det behövs.
- 6 : Klicka på 'Acceptera' för att använda dessa offseter för din nya klass, eller på 'Avbryt' för att avbryta.

VARNING :

För att lärfunktionen skall fungera måste du se till att :

- * ALLA VALDA FILER TILLHÖR SAMMA KLASS.
- * ALLA VALDA FILER EJ ÄR KOMPRIMERADE.

Det är allt !!! Enkelt, eller hur ?

1.8 Vad är en Klass ?

En Klass är en filfamilj. Tex C-filer kan ses som en Klass, låt oss kalla den C-Klassen.

Med ClassActionPrefs kan du skapa hur många Klasser du vill, så länge du kan förklara hur de skall kännas igen.

För att förklara hur en Klass skall kännas igen finns det två mtoder : matchande filnamn och filinnehåll.

Matchande filnamn används för att beskriva en Klass mha dess filnamn.
Filinnehåll används för att beskriva en Klass mha dess innehåll.

Det finns två inbyggda Klasser som du inte kan ta bort, de visas med vit färg i KlassListan.

Den första heter "Okänd Klass", men du kan döpa om den.
Denna Klass beskriver alla filer som ClassAction inte kan känna igen.

Den andra heter "Generella Handlingar" som du inte kan döpa om.
Denna Klass beskriver Handlingar som kommer att visas för ALLA andra Klasser...

Intressant : Om du vill ha en kopierings-Handling för alla Klasser kan du skapa en sådan för varje Klass som du definierar, detta tar lååååång tid och är tråkigt.
Ett bättre sätt att genomföra detta är att skapa denna Handling under "Generella Handlingar"-Klassen. Detta medför att kopierings-Klassen visas i alla Klasser.

De generella Handlingarna visas med vit färg i ClassAction's HandlingarLista och kan endast ses i ClassActionfönstret.
De kommer inte att visas när ClassAction är en AppIkon, aller

via ARexx-kommandot Load.

1.9 Skapa en ny Klass

Klicka på 'Addera'-knappen i Klassdelen för att skapa en ny klass.

En Klass har tre egenskaper :

- Ett namn
- Ett matchande filnamn
- Offseter

- o Namnet är helt enkelt Klassens namn. Det är helt upp till dig hur du vill döpa den.
VARNING: Ett klassnamn måste vara unikt.
- o Matchande filnamnet är ett ordinärt AmigaDOS-uttryck :

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

(Läs AmigaDOS-manualen där alla wildcards beskrivs.)

VARNING: Använd inte *, utan använd #? istället.

Matchnamnet är ej känsligt för stora och små tecken, därför matchar toto.C med #?.c

Om du skapar en Klass som använder matchande namn, måste du försäkra dig om att definitionen är bra.

Exempel : Om du definierar GIF-Klassen med matchnamnet #?.gif kommer alla filer som slutar med .gif i filnamnet att kännas igen som GIF-filer.

Men är du säker på att ALLA .gif-filer är GIF-bilder, eller att alla dina GIF-bilder slutar med .gif ?

Därför bör du endast utnyttja matchande namn i två situationer :

- Det matchande namnet är helt unikt för Klassen.
(Tex: #?.info är tillräckligt bra för en IkonKlass)
- När du inte har något annat val.
(Tex: Hur skall en C-källkod kännas igen, förutom med #?.c)

- o I andra fall skall du använda Offseter.

En offset är en plats i en file där vi kan hitta någonting som vi känner igen.

Tex, GIF-bilder börjar alltid med texten 'GIF' vid offset 0.

Det finns tre uttryckssätt för att definiera offseter :

=====
Metod #1 : Offset,HexSträng

Offset är ett DECIMALT tal som talar om offseten.
HexSträng är en hexadecimal sträng som skall återfinnas vid denna offset.

Exempel 1 : 0,4f4a Betyder att filen måste börja med två bytes, \$4f och \$4a på position 0.

Exempel 2 : 9,448b3c Betyder att vid byte #9, skall vi finna \$44 \$8b \$3c.

=====
Metod #2 : Offset,'Sträng'

Offset är ett DECIMALT tal som talar om offseten.
Sträng är en ASCII sträng som skall återfinnas vid offseten.

Exempel 1 : 0,'GIF' Betyder att filen måste börja med texten 'GIF'

Exempel 2 : 9,'FuBar' Betyder att vid byte #9, skall vi finna texten 'FuBar'.

=====
Metod #3 : Offset,"Sträng"

Offset är ett DECIMALT tal som talar om offseten.
Sträng är en ASCII sträng som skall återfinnas vid offseten.

Notera skillnaden mot förgående uttryck, här används " för att definiera strängen, i Metod #2 använde vi '.

Detta är samma sak som i Metod #2, men här tar strängjämförelsen INTE hänsyn till stora och små bokstäver.

Tex: En AmigaGuide-fil börjar alltid med texten :
@database med små eller stora bokstäver.

Om du skulle använda Metod #2 för att känna igen en AmigaGuide-fil genom att ange 0,'@database', kan inte ClassAction känna igen en fil som börjar med texten @DATABASE som en AmigaGuide-fil.

Det fungerar däremot om offseten är definierad enligt
Metod #3 : 0,"@database"

=====

Du kan definiera upp till 5 offseter för att beskriva en Klass.
En fil anses tillhöra en Klass om alla offseter stämmer.

Ex : Om Klassen X är definierad så här :

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

Kommer alla filer som börjar med 4a8b6c OCH har \$14 vid byte #58
att anses tillhöra Klassen X.

För att definiera flera offsets, klicka på cycle gadgeten
'Offset #' för att aktivera nästa offset.

Notis : ASCII-Klassen

Det finns ett inbyggt offsetkommando : ASCII[]

Om du använder detta kommando i offset #1 (Offset#1=ASCII[]),
kommer detta matcha ASCII-filer.
ClassAction kommer dock att pröva detta när allt har misslyckats.
Tack vare detta kommer AmigaGuide-filer (som är ASCII-filer) inte
att klassas som ASCII om du redan har definierat en AmigaGuide-
Klass.

Normalt sett behöver denna Klass inte används pga att jag har
skapat en standard Prefs-fil där klassen ASCII finns definierad
mha detta kommando.

1.10 Skapa en ny Handling

När en Klass har skapats bör du definiera en Handling för den.
Varje Klass kan så många Handlingar som du vill.

Klicka på 'Addera'-knappen vid HandlingsDelen för att lägga till
en Handling.

En Handling har fem beståndsdelar :

- Ett namn
 - En exekveringsmod
 - En stackstorlek (Endast vid exekvering från CLI)
 - En fördröjning (Endast vid exekvering från CLI)
 - Ett exekveringskommando
-
- o Namn är Handlingens namn.
 - o Exekveringsmod kan vara
 Cli

```

'
WB
'
Ej Cli
eller
ARexx
.

```

- o Exekveringskommando är ett giltig AmigaDOS-kommando som kan innehålla parametrar. DU BÖR alltid ange den fullständiga sökvägen för programmen.

Exempel : Ange C:Copy istället för Copy.

```

Du kan ange ClassAction's inbyggda
Argument
och
Kommandon
.

```

Knappar

- o 'U' Upp och 'D' Ned knapparna används för att sortera handlingarna.
- o 'Ladda'-knappen frågar efter en exekverbar fil (program).
- o 'Komm'-knappen öppnaren dialog som visar alla tillgängliga argument och kommandon för ett exekveringskommando.
- o Med 'Kopiera'-knappen kan du kopiera alla Handlingar från en Klass till en annan. Klicka på 'Kopiera'-knappen och välj den Klass vars Handlingar skall kopieras från. Detta är bra att ha när samma Handlingar skall användas för flera Klasser.

1.11 CLI mod

**** CLI mod ****

Om 'CLI' har valts, kommer den valda Handlingen att starta ett CLI och sätta stackstorleken till det angivna stackvärdet (normalt 4096).

Exekveringsmodens CLI kommer endast att öppna ett CLI om det är nödvändigt (dvs om programmet skriver ut någonting till fönstret).

Du kan ange fördröjningen för CLI:

Om fördröjningen är negativ (tex Delay = -1) kommer CLI-fönstret att finnas kvar ända tills du stänger det för hand med stäng-knappen i det övre, vänstra hörnet av fönstret.

Om fördröjningen är noll (Delay = 0) kommer CLI-fönstret att stängas

automatiskt när programmet har avslutats.

Om fördräjningen är positiv (Delay = n, där n > 0) kommer CLI-fönstret att vänta n sekunder innan det stänger sig självt. Du kan dock tvinga fönstret att försvinna mha stängknappen.

Storleken på CLI-fönstret kan bestämmas med verktygstypen CLISIZE. (Den gamla verktygstypen OUTPUT finns ej längre).

1.12 WB mod

*** WB mod ***

Om 'WB' har valts kommer inget CLI-fönster att öppnas. Istället simulerar ClassAction att Workbench har startat programmet. Programmet (verktyget) kommer att startas med argumenten som finns angivna i dess ikon.

Varning: Denna mod är endast tillgänglig för program som har en ikon.

1.13 Ej CLI mod

*** Ej CLI mod ***

Om 'Ej Cli' är valt kommer inget CLI-fönster att öppnas, även om programmet försöker skriva ut någonting. Programmet kommer fortfarande tro att det startades från ett CLI.

1.14 ARexx mod

*** AREXX mod ***

Om 'ARexx' är valt kommer programmet Rx att startas med det angivna (ARexx) programmet. Programmet MÅSTE vara ett ARexx-program. Givetvis måste REXXMaster vara aktiverad och Rx finnas i biblioteket SYS:rexxc för att detta skall fungera.

1.15 Argument

För närvarande finns åtta argument tillgängliga :

De första fyra kommandona anges med gemena bokstäver, [f] [s] [b] [x] och de sätter resultatet inom citationstecken.

[f] : Fullständig sökväg för filen med citationstecken.

[s] : Fullständig sökväg för filen utan suffix med citationstecken.

[b] : Filnamnet för filen med citationstecken.

[x] : Filnamnet för filen utan suffix med citationstecken.

De resterande fyra kommandona resulterar i samma sak som de fyra först men utan citationstecken.

[F] : Fullständig sökväg för filen.
[S] : Fullständig sökväg för filen utan suffix.
[B] : Filnamnet för filen.
[X] : Filnamnet för filen utan suffix.

Exempel : Antag att du väljer filen 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
```

Exempel: Anta att du har valt filen ram:main.c

```
* Kommandoraden  
    c:copy [f] [F].bak  
kommer att ersättas med :  
    c:copy "ram:main.c" ram:main.c.bak
```

```
* Kommandoraden  
    c:copy [f] [S].bak  
kommer att ersättas med :  
    c:copy "ram:main.c" ram:main.bak
```

1.16 Kommandon

För närvarande finns det fem möjliga dialogkommandon :

```
REQD[text] : Frågar efter ett bibliotek.  
REQF[text] : Frågar efter ett filnamn.  
REQV[text] : Frågar efter en volym (disk).  
REQT[text] : Frågar efter en text.  
SURE[text] : Ber användaren att bekräfta.
```

=====

REQs Kommandon

REQs kommandon öppnar en Reqtools-dialog med titeln [text]. Detta är användbart när du behöver interaktiva kommandorader.

Exempel :

```
bin:lha x [f] to REQD[Välj ett bibliotek som skall packas upp]
```

Detta exempel öppnar en dialog som låter användaren välja destinationsbiblioteket. Därefter packas det valda arkivet [f] upp i det valda biblioteket.

REQF[] fungerar likadant, förutom att det frågar efter en fil.

Exempel :

```
c:dir [f] > REQF[Välj en fil]
```

REQV[] ber användaren att välja en volym (disk).

REQT[] frågar efter en text. Detta är användbart när ett program behöver ett eller flera argument angivna:

```
c:cpu REQT[Ange argument för Cpu]
```

Varning :

REQD, REQF och REQV kan vara inkompatibla med AppIkonen när du använder program som tex MagicMenu.

Om du får 'gurus' regelbundet när REQF, REQD och/eller REQV används samtidigt som du använder AppIkonen, försök hitta programmet som stör ClassAction, eller sätt verktygstypen REQBUG till YES.

Om du sätter denna verktygstyp till YES, kommer REQD, REQF och REQV som används från AppIkonen att automatiskt göras om till ett REQT-kommando (säkrare kommando).

=====

SURE-kommandot:

SURE[text]-kommandot öppnar en dialog med texten [text] och med två knappar, Ja / Nej.

Om användaren väljer Nej, exekveras inte programmet.

Om användaren väljer Ja, kommer ClassAction att exekvera kommandot på exekveringsraden som står till HÖGER om Sure-kommandot.

Exempel :

```
SURE[Vill du verkligen radera denna fil ?]C>Delete [f]
```

Detta exempel kommer att öppna en dialog som ber användaren

att svara Ja eller Nej på frågan "Vill du verkligen radera denna fil?".
Om användaren svara Nej kommer nget att hända ; om användaren svara Ja kommer komandot C>Delete [f] att exekveras.

=====

Givetvis går det att kombinera hur många kommandon / argument som helst i en exekveringsrad.

Exempel :

```
SURE[Döp om filen ?]c:rename [f] REQF[Ange det nya namnet]
```

1.17 Framtida förbättringar

Vad jag vill lägga till detta program i en framtida version :

- o Klassfamiljkoncept :
Tex: Bild-Familjen innehåller GIF, IFF, TARGA, JPEG...
och möjligheten att välja ett Familjefilter i fildialogen.
Med detta är det möjligt att filtrera bilder, ljud...
- o Obegränsat antal sökvägsknappar.
- o Se till så att ClassActionPrefs använder WB-typsnittet.
- o Och givetvis allt som du ber om :)

1.18 Om

Jag skulle vilja tacka följande personer :

- o Mireille (för hennes tålamod...)
- o Philippe Thomas (för förslag, hjälp, betatestning, den franska guide-filen och UTT som har använts för Installationsprogrammet)

Phil, om du läser detta, klicka
här
 - o Richier Pierre (för MagicWB-ikonerna för ClassAction)
- o Jean Michel och Georges (för betatestning på en A4000/40)
- o Obvious Implementations Corp (för Dice C Pro)
- o Nico Francois för ReqTools library
- o Georg Hörmann för det *UNDERBARA* xfdmaster library

- o Alla användare som har kontaktat mig angående buggar eller förslag till uppgraderingar.
- o Alla registrerade användare.

1.19 Författaren

Du kan kontakta mig på följande adress :

Gasmi Salim
6, rue des Hirondelles

67380 Lingolsheim
France

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

1.20 Using ClassAction

Att använda ClassAction är VÄLDIGT enkelt.

Bara välj en fil med dialogen.
Du kan gå till moderbiblioteket mha knappen 'Moder'
eller textraden <Parent> i dialogrutan.
Högra musknappen visar volymer (diskar) och tilldelningar.

Efter filvalet syns filklassen och dess handlingar i den högra listan.
Välj den handling som du önskar skall utföras...

Om du dubbelklickar på en fil kommer den första handlingen i listan att startas.

Du kan Zooma fönstret med den lilla Zoom-knappen i det övre, högra hörnet i fönstrets titelrad.

För att avsluta, välj "Avsluta"-knappen.

För att göra om fönstret till en AppIkon, stäng fönstret.
Dubbelklicka AppIkonen för att få tillbaka fönstret.

När ClassAction körs som en AppIkon kan du släppa andra ikoner över AppIkonen så att dessa filtyper identifieras.
Om Klassen bara har en enda Handling definierad kommer denna att utföras.
Annars öppnas ett litet fönster som visar en lista över de Handlingar som finns definierade för Klassen ifråga. Ur detta fönster behöver du bara välja en Handling så startas den.

Det var allt, enkelt va ???

1.21 Teknisk information

ClassAction är till 100 % kodad med DICE C 3.0

Övrig information :

Inställningsfilen är en ASCII-fil med namnet ENVARC:ClassAction.prefs
De generella Handlingarna sparas i filen ENVARC:ClassAction_Gen.prefs

ClassAction skapar ett extraprogram som heter ClassAction_RunTask
och sparas i T:

Detta program används för att starta WB-program.

Hastigheten i Fildialogen beror på den använda sorterings-
algoritmen (rekursiv trädsortering som kallas "Tri du Chat")

Programbibliotekinfo :

Utnyttjade ROM-bibliotek :

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

Uttyttjade DISK bibliotek som behövs :

rexsyslib.library	V39+
commodities.library	V37+
asl.library	V39+
icon.library	V37+
reqtools.library	V38+

DISK bibliotek som utnyttjas om de kan hittas :

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

Hur ClassAction identifierar en Klass :

- 1- Testa om filnamnet matchar filmatchnamnet för en Klass.
- 2- Testa om filen matchar Offseterna för en Klass.
- 3- Dekomprimera filen mha xfdmaster.library.
- 4- Testa om den uppackade filer matchar Offseterna för an Klass.
- 5- Testa om filen är en ASCII-fil (om ASCII-kommandot existerar).

Om allt misslyckas anses filen vara av "Okänd Klass".

1.22 Installation

För att installera ClassAction :

Normalt skall du använda Installer-skriptfilen som finns i detta arkiv och följa instruktionerna.

Men om du inte har detta (eller Installerprogrammet), gör så här :

- Kopiera ClassAction, ClassActionPrefs och dess ikoner var du vill.
- Kopiera .prefs-filerna till ENVARC:
- Kopiera ClassAction.guide vart du vill.

Detta är allt...

1.23 Registrering

Om du läser dessa rader undrar du hur du skall bli en ←
registrerad
ClassAction-användare.

Låt mig förklara varför du skall registrera dig :

Först av allt för att stödja den bästa dator som någonsin har tillverkats, därför att Amigans framtid beror på hur den framtida tillgången till mjukvara ; så när du stödjer en programmerare för hans arbete, stödjer du din dator och hans framtid !!!!

Dessutom därför att jag spenderar all min fritid för att försöka göra Shareware-program.

Om du använder dem, varför inte skicka mig registreringsavgiften ?
Detta får mig att fortsätta att göra Shareware's.

Shareware-versionen är 100% användbar, inget har gjorts otillgängligt.
Men om du verkligen gillar det, REGISTRERA DIG !!

Registreringsavgiften är (beroende på din lokala valuta) :

10 US\$ eller 20 DM eller 50 FF.

Sänd avgiften till min
adress

så får du den senaste registrerade versionen av ClassAction.

Glöm inte att ange HELA din adress (även nationalitet).

Tack på förhand för ditt stöd !

Hälsningar,
Salim

1.24 Lagliga ting

Copyright

ClassAction och ClassActionPrefs är Copyright © 1994-1995
av Gasmi Salim.

ClassAction är ett shareware-program. Arkivet får inte förändras
på något sätt och får inte användas för kommersiella avsikter
utan föregående skriftligt tillstånd av författaren.
Copyright-meddelande skall bevaras.

Garanti

Inget ansvar accepteras för eventuell skada som kan uppstå pga att
detta program har använts.
All användning sker på egen risk. Mjukvaran levereras "som den är"
utan någon ingående garanti för lämpligheten eller noggrannheten
i programmet och dokumentationen.
Dokumentationen antages vara korrekt, men författaren reserverar sig
för rätten att uppdatera mjukvaran och/eller dokumentationen
utan förestående notis.

1.25 Historik

ClassAction Historik V 2.8 (c) Salim Gasmi

950925 : V2.8

- ClassAction stöder nu Commodities FULLSTÄNDIGT.
Har dessutom en Hotkey för att visa/gömma ClassAction.
 - Verktystypen APPSTART kan sättas till HIDE (APPSTART=HIDE)
om du vill att ClassAction skall startas undangömt.
-

- Verktygstypen CX_HOTKEY införd.
- För att tillåta användning av publika skärmar har verktygstypen PUBSCREEN lagts till.
- ClassAction's interna Filväljare returnerade inte allt allokerat minne, detta har fixats till.

950911 : V2.75

- REQD, REQF och REQV-kommandona var i verkligheten inte inkompatibla med AppIkonen, utan inkompatibla med en del program såsom MagicMenu (otur att jag använder MagicMenu...)

Flera användare klagade på den automatiska omkopplingen till ett REQT-kommando även om de inte använde ett inkompatibelt program.

Jag har lagt till en verktygstyp (REQBUG) som låter användaren välja om auto-omkoppling till REQT skall ske eller ej.

950903 : V2.7

- Kommandona [b],[x],[B],[X],[F] och [S] har lagts till.
- Arexx-kommandot Action har lagts till.
- Några buggar i grafikanropen har tagits bort.
- En bugg angående flera ikoner som släpps över AppIkonen har tagits bort.
- Kommandona REQD, REQF och REQV kopplas om till ett REQT-kommando när dessa används från AppIkonen eller Arexx, de är endast kompatibla med fönsterläget.

950828 : V2.6

- Storleken på ClassAction's fönster går nu att ändra på.
 - ClassAction och ClassActionPrefs använder ReqTools.library.
 - REQV-kommandot som frågar efter en volym (disk) har lagts till.
 - REQT-kommandot som frågar efter en text har lagts till.
 - WINX- och WINY-verktygstyperna har lagts till.
 - Lär-dialogen har fått en Alla-knapp.
 - Vald fil finns nu i alla REQs-dialoger.
-

- ClassAction använder numera en Nyckelfil för registrerade versioner, filen placeras i S:.

950717 : V2.5

- ClassAction och ClassActionPrefs är nu lokaliserade och en Fransk katalog finns med i arkivet.
- Lär-funktion har lagts till i ClassActionPrefs.
- För att välja den första Handlingen för en fil, måste du numera dubbelklicka på den istället för att välja om filen.
- Generella Handlingar är nu synkroniserade och läser om biblioteket.
- Verktygstypen AUTOSELECT används ej längre.
- REQs-dialoger öppnas nu i nuvarande bibliotek.
- Några mindre förbättringar har gjorts.

950612 : V2.1

- "Generella Handlingar" har lagts till som en inbyggd Klass.
 - Exekveringskommandot SURE[] har lagts till.
 - Offsetkommandot ASCII[] som känner igen ASCII-filer har införts.
 - Följande Arexkommandon är nya: AppIconify, Show, Status, GetClass.
 - Exekverande program har fått ett ändrat system, nu använder jag SystemTags(). Vi behöver inga tmp-filer längre.
 - Du kan nu ange en fördröjning för CLI-exekveringar.
 - 'sträng' och "sträng"-definitionerna har lagts till i Offsets.
 - Verktygstypen OUTPUT är numera förlegad och används inte längre. Numera används verktygstypen CLISIZE istället.
 - Handlingsdialogen har nu en bättre storlek. Den dyker upp under muspekaren och på den främsta publika skärmen.
 - Knapparna "Använd" och "Spara" har bytt plats och "Avbryt" har lagts till i ClassActionPrefs för att följa Amiga-inställningarnas utseende.
"Om"-knappen har flyttats till övre, högra hörnet och döpts om till "?".
 - ClassActionPrefs Cycle gadgets-rutiner var inte 100% systemvänliga och vissa patchar som CycleToMenu fungerade inte med ClassActionPrefs; Detta är fixat nu.
-

- Igenkänningskoden och Info-rutinen har förbättrats: de är upp till 400% snabbare.
- Fel med Listview hilight-färg her elimierats.
- ClassAction låser inte WorkBench-skärmen längre när det är AppIkonifierat.
- En otäck bugg har upptäckts och eliminerats i ClassActionPrefs.
- Högra musknappen visar tilldelningar endast om muspekaren finns i dialogen och ett förnyat höger musknapp-tryck återhämtar biblioteket.
- Nu memorerar ClassAction fönstrets position.

950523 : V2.00 (Större uppdatering)

- ClassAction har numera en AppIkon.
- ClassAction är nu en sk Commodity.
- ClassAction har fått en Arexx-port.
- ClassAction använder numera standard WB-typsnittet.
- Exekveringskommandot "Arexx" har lagts till.
- Olika färger för bibliotek och filer.
- Upp- och Ned-knappar har lagts till i ClassActionPrefs.
- "Använd"-knapp har lagts till i ClassActionPrefs.
- Klasserna sorteras i ClassActionPref's Listview.
- Verktygstyperna APPSTART, ICONNAME, ICONX, ICONY, CX_PRIORITY, WBFONT, OUTPUT och ICONFILE har tillkommit.
- När fönstret har ikonifierats har det rätt höjd med hänsyn till skärmens standardtypsnitt.
- Nytt Spara-format (CASF20).
- Suffix/Prefix-knappen borttagen. Ersatt med MatchNamn-knappen som accepterar alla sk Wildcards.
- Konfigurationsfilen flyttad till ENVARC:
- Ett installationsskript finns distribuerat i arkivet.
- En del kodoptimering har utförts.

950505 : V1.43

- Dialogkoden har rensats upp, 5% snabbare numera.
- ClassAction letar efter sitt namn mha WBStartUp-strukturen används och kan dessutom döpas om.

950502 : V1.42

- "Ej CLI" exekvering har lagts till i ClassActionPrefs.
- "" läggs alltid till i filnamnen även om det inte behövs, detta förenklar i Arexx-program.

950406 : V 1.4

- Verktygstypen HEIGHT har lagts till.
- En del kodoptimering har utförts.

950222 : V 1.31

- Om du klickar två gånger på samma fil kommer den första Handlingen att startas (detta går snabbare än att välja den första Handlingen för hand).
- Denna version är numera Shareware och du måste registrera dig för att få den registrerade versionen.

950115 : V 1.3

- De interaktiva kommandona REQD[] och REQF[] har lagts till.
- "Kopiera"-knapp har tillkommit i ClassActionPrefs.
- Mindre förbättringar har gjorts.
- Beta testare har rapporterat att denna version är riktigt stabil.

941125 : V 1.22

- Första officiella utgåvan.
 - Knappen "Info" har tillkommit.
 - ClassAction gjorde Locking() på bibliotek som det lästa utan att Unlocking() desamma.... *FIXAT*
 - Kodoptimering.
 - Några övriga, mindre buggar har eliminerats.
-

941108 : V 1.21

- Programmet kraschade när flyppyenheterna var tomma. *FIXAT*
- Volym/Namn-bugg fixad.
- <..>-text borttagen när volymroten var nådd.

941107 : V 1.2

- Nytt gränssnitt, jag har introducerat min egen, snabba fildialog.
- Verktygstyperna STARTDIR & DRIVE1 tom DRIVE11 har tillkommit.

941101 : V 1.1

- Använder numera xfdmaster.library för att identifiera och dekomprimera filer.
- Konfiguration mha verktygstyper har lagts till (DECRUNCH, AUTOSELECT).
- "Okänd Klass" är nu en inbyggd Klass med obegränsat antal Handlingar.
- Nytt Spara-format (CASF11).

941016 : V 1.0

- Nytt Spara-format (CASF10).
- Fönstret har nu en Zoom-knapp.
- Flera Klass-definitioner har lagts till.

941010 : Beta Version

- tmp fil-bugg borttagen.
- Inkrementering av Offset-bugg borttagen.
- Använder asl.library för fildialogen.

941001 : Alpha Version

1.26 Arexx kommandon

ClassAction har en Arexx-port som heter ClassAction.01

Här följer en lista över alla Arexx-kommandon :

=====

Quit

Avslutar ClassAction...

=====

Use

Tvingar ClassAction att läsa in inställningsfilen.

=====

Ver

Returnerar ClassAction's version.

=====

Status

Returnerar ClassAction's nuvarande status.

0 : ClassAction
0 : ClassAction är AppIkonifierad.
1 : ClassAction körs som fönster.

=====

AppIconify

Tvingar ClassAction att bli en AppIkon.

Om ClassAction redan är en AppIkon händer ingenting.

=====

Show

Tvingar ClassAction att köra i fönsterläge.

Om ClassAction redan visar sitt fönster, händer ingenting.

=====

Load <filnamn>

ClassAction försöker läsa filen <filnamn>, och därefter visa handlingsdialogen så att användaren kan välja en Handling.

Om filen inte finns, händer ingenting.

=====

GetClass <filnamN>

ClassAction försöker läsa filen <filnamn> och returnerar filens Klass.

Om filen inte finns, händer ingenting.

=====

Action <filnamn> <Handlingsmönster>

ClassAction exekverar första matchande Handlingen för <filnamn>.

Exempel : Action ram:toto.lha extr

Kör första Handlingen med namnet som matchar strängen 'extr' på filen ram:tot.lha

Mönstret är användbart därför att du inte behöver ge det exakta Handlingsnamnet, bara en del av namnet räcker.

1.27 Vad du skall göra efter installationen

Efter installationen skall du ha en fungerande ClassAction med flera Klasser definierade, men med väldigt få angivna Handlingar (de flesta av Klasserna har inga Handlingar definierade alls).

Det är upp till dig att definiera Handlingar med utgångspunkt från din systemkonfiguration och de program som du föredrar att använda.

Därför skall du starta ClassActionPrefs och konfigurera alla Handlingar (och Klasser) till din bekvämlighet...

Om du inte vet hur du skall konfigurera Handlingar och Klasser, läs denna guide :)).

Efter att Klasser och Handlingar är konfigurerade, ställ in ClassAction's uppträdande mha dess verktygstyper.

Det kan ta lång tid att skapa en 'trevlig' konfiguration. Men när det väl är gjort så är det BRA !!!

Ok, läs nu denna guide och Lycka Till.

1.28 Meddelande till Philippe THOMAS

Salut Philippe !!!

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

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

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

Franchement si ClassAction commence à être cool, c'est beaucoup grâce à toi, je crois que j'aurais eu la flemme de le paufinner autant si tu n'étais pas là.

Bref, ce programme est aussi un peu le tien.

Okay Phil, à la prochaine.

Salim.

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

1.29 Kända buggar

Flera användare har frågat mig om ett användargränssnitt (GUI) vars storlek kan ändras. Jag har gjort mitt bästa för att införa detta i denna V2.6+ version, men det är en ganska knepig uppgift med GatTools.

Väldigt sällan, när du ändrar storlek på fönstret, 'hänger sig' fönstrets uppdatering och du måste stänga fönstret för att 'väcka' det.

Jag vet faktiskt inte varför eftersom jag aldrig har sett detta problem själv. Beta-testarna har rapporterat detta problem och de noterade det endast två gånger under en månads testkörningar.

Hursomhelst, jag gör mitt bästa för att hitta och eliminera denna bugg, men kommer inte att förneka många användare ett GUI som kan ändras i storlek pga av en bugg som syns två ggr per månad.

Visaa program som MagicMenu, gör att REQs-kommandona kraschar när dessa används från AppIkonen.

Du kan använda verktygstypen REQBUG för att stänga av REQs-kommandona.