

AppMenu_D

COLLABORATORS

	<i>TITLE :</i> AppMenu_D		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		November 2, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AppMenu_D	1
1.1	AppMenu Guide	1
1.2	Copyright	1
1.3	Einführung	2
1.4	Installieren und starten	3
1.5	Benutzen	3
1.6	Symbol: Menüpunkte	5
1.7	Symbol: Befehl	5
1.8	Symbol: '>' - Befehl wählen	6
1.9	Symbol: Arg	6
1.10	Symbol: Neu	6
1.11	Symbol: Lösch	6
1.12	Symbol: Speichern	7
1.13	Symbol: Benutzen	7
1.14	Menü: Projekt/beenden; Symbol: Abbrechen	7
1.15	Menü: Projekt/öffnen ...	7
1.16	Menü: Projekt/speichern als ...	7
1.17	Menu: Vorgaben/Ausschneiden	8
1.18	Menü: Vorgaben/Kopieren	8
1.19	Menü: Vorgaben/Einfügen	8
1.20	Menü: Vorgaben/Bewegen rauf, Vorgaben/Bewegen runter	8
1.21	Menü: Vorgaben/auf zuletzt gespeichertes	8
1.22	Menü: Vorgaben/auf vorherigen Stand	8
1.23	Menü: Optionen/Piktogramme erzeugen?	9
1.24	Fehler-Meldungen	9
1.25	Autor und Programme	10
1.26	Geschichte and was neues	10

Chapter 1

AppMenu_D

1.1 AppMenu Guide

AppMenu - Version 2.4 (8.10.95)

Copyright © 1993-1995 Marius Chwalczyk.
Alle Rechte vorbehalten.

Copyright

Einführung

Installieren und starten

Benutzen

Fehler-Meldungen

Autor und Programme

Geschichte und was neues

1.2 Copyright

Das Paket "AppMenu - Version 2.4" steht unter Copyright © 1993-1995 von Marius Chwalczyk. Alle Rechte sind vorbehalten.

Es ist VERBOTEN das Paket zu verkaufen oder sonstwie weiterzugeben. Das Paket kann nur unter folgenden Bedingungen frei weitergegeben werden:

1. nur minimale Gebühr (Kopierkosten, Datenträgerpreis) sind bei der Weitergabe erlaubt.
 2. alle hier aufgeführten Dateien müssen in ihrem originellen Zustand, ohne irgendwelche Änderungen enthalten sein:
-

AppMenu	11628	-	der Voreinsteller (WB 3.x)
AppMenuCx	4824	-	das Programm (WB 3.x)
Copyright-LiesMich	1856	-	diese Datei
Copyright-ReadMe	1607	-	diese Datei in englisch
AppMenu_D.guide	22155	-	deutsche Dokumentation
AppMenu.guide	17369	-	englische Dokumentation
Workbench_2.x	(Dir)	-	
AppMenu	12352	-	> Versionen für Workbench 2.x
AppMenuCx	5480	-	
AppMenu.info	1052	-	
AppMenuCx.info	860	-	
Copyright-LiesMich.info	835	-	
Copyright-ReadMe.info	835	-	> Piktogramm-Dateien
AppMenu_D.guide.info	840	-	
AppMenu.guide.info	840	-	
Workbench_2.x.info	628	-	
AppMenu.info	1052	-	
AppMenuCx.info	860	-	

3. die Dateien können gepackt werden (z. B. mit LhA oder Lharc).
Es ist aber kein Crunchen (das Packen mit beiegehaltener
Lauffähigkeit) der ausführbaren Dateien erlaubt.

Falls das Paket in PublicDomain-Serien aufgenommen wird, bitte
ich (

der Autor

) um eine Information darüber. Für eine beigelegte,
kostenfreie Kopie der Ausgabe danke ich voraus!

Das Programm ist "wie es ist", jede Benutzung ist auf eigene Risiko.

1.3 Einführung

Das Paket AppMenu enthält zwei Programme. Beide laufen nur ↔
unter Release 3

(V39) und späteren Versionen von AmigaOS. Es gibt auch Versionen für Release 2
(V37/38) in der Schublade "Workbench_2.x".

Das eine

AppMenuCx

ist ein Commodity mit dem eigene Menüpunkte in das
Hilfsmittel-Menü von Workbench eintragen werden können. Wird eins dieser
Punkte angewählt, führt AppMenuCx ein bestimmtes Befehl aus. Der Befehl wird
als eigener, asynchroner Prozeß gestartet. Seine Argumente können durch
Piktogramme oder im Dateiauswahlfenster angegeben werden.

Alle Menüpunkte, die Befehle, sowie Art der Argumentenübergabe sind mit
Hilfe des zweiten Programms

AppMenu

frei definierbar.

Das Programm funktioniert wie ein System-Voreinsteller der Workbench.

1.4 Installieren und starten

Kopieren Sie beide Programme
AppMenuCx und AppMenu
in Verzeichnisse
ihrer Wahl. Das erste am besten in den WBStartup-Ordner, das zweite in den
Prefs-Ordner.

Es werden zwei Vorgabendateien (wie bei den System-Voreinstellern) benutzt:
"ENV:AppMenu/AppMenu.prefs" und "ENVARC:AppMenu/AppMenu.prefs". Die beiden
Verzeichnisse müssen zuerst mit den Befehlen: "MakeDir ENV:AppMenu" und
"MakeDir ENVARC:AppMenu" erstellt werden.

Die Programme nutzen die Sprach-Lokalisierung ab der Release 2.1 von
AmigaOS. Falls eine der beiliegenden Katalogdateien der gewählten Sprache
entspricht, kann sie in eines der Verzeichnisse:

"LOCALE:Catalogs/<Name der Sprache>/" oder
"<Verzeichnis des Programms>/Catalogs/<Name der Sprache>/"

kopiert werden.

Beide Programme können über Workbench oder Shell gestartet werden.

AppMenuCx ist ein Commodity und benutzt nur ein Shell-Argument, bzw. Piktogramm-Merkmal: CX_PRIORITY=<n>. Schablone ist CX_PRIORITY/K/N. Vorgegeben ist 0. Genaueres siehe in dem Workbench Handbuch unter Commodities.

AppMenu versteht folgende Shell-Argumente: FROM=<Dateiname>, EDIT, USE, SAVE und CLIPUNIT=<Clip Unit Num>. Schablone ist FROM,EDIT/S,USE/S,SAVE/S, CLIPUNIT/K/N. '<Dateiname>' bezeichnet die Vorgabendatei (normalerweise "ENVARC:AppMenu/AppMenu.prefs"). Vorgegeben '<Clip Unit Num>' ist 0 (erster Clipboard). Die Bedeutung der Argumente ist analog wie bei den Voreinstellern der Workbench.

Außerdem gibt es folgende Merkmale des Piktogramms:

CLIPUNIT=<Clip Unit Num> sehe oben.

SAVEICONS=<NO|YES> bestimmt den Zustand des
Piktogramme erzeugen?
Menü-

punktes. Vorgegeben ist YES. Beim Shell-Start zeigt das Menü NO.

SCREENFONT=<NO|YES> bestimmt, ob AppMenu den voreingestellten Fenster-Zeichensatz statt den "Topaz/8" benutzt. Vorgabe ist YES. Es wird immer benutzt, auch beim Shell-Start.

1.5 Benutzen

Mit AppMenu wird eine Datei mit der Beschreibung der einzelnen ↔
Menüpunkte
erstellt. AppMenuCx benutzt diese Vorgabendatei, um die Punkte in dem
Hilfsmittel-Menü zu erstellen und ihre Befehle zu starten.

Das Fenster von AppMenu enthält eine Rollliste mit ihrem Rollbalken und


~~~~~

auf zuletzt gespeichertes 'L'

auf vorherigen Stand 'S'  
Optionen

Piktogramme erzeugen? 'I'

AppMenuCx hat kein Fenster. Es kann aber wie jeder Commodity ←  
mit dem

Programm "Exchange" kontrolliert werden. AppMenuCx benutzt über Notifikation  
immer die aktuelle "ENV:AppMenu/AppMenu.prefs" Datei.

## 1.6 Symbol: Menüpunkte

Die Rolliste zeigt die Liste der Menüpunkte. Wenn Sie ein ←  
Menüpunkt ändern

wollen, wählen Sie es aus der Rolliste aus. Sie können dann, den Text des

Befehls

modifizieren. Drücken Sie die Eingabe- oder die Tab-Taste, oder

ändern Sie das

Arg

Blättersymbol, um das Textfeld unter der Rolliste

zu aktivieren. Hier können Sie den Text des Menüpunktes ändern. Mit der  
Eingabetaste wird das Menüpunkt in die Liste zurückgenommen.

## 1.7 Symbol: Befehl

Geben Sie hier den Namen des Befehls und seine Argumente ein, ←  
das Sie

starten wollen. Sie müssen den ganzen Pfad des Befehls angeben, wenn AppMenu  
über Workbench gestartet wird, weil in diesem Fall das Programm nur den Pfad  
"C:" kennt. Sie können das Problem umgehen, indem Sie folgende Zeile in die  
"S:User-startup" zufügen: "ASSIGN C: <Pfad\_des\_Befehls> <...> ADD".

Die zusätzliche

Argumente

werden am Ende des Befehls hinzugefügt. Braucht

das Programm sie aber an einer bestimmten Stelle, kann die Stelle wie in dem  
"Alias" Befehl durch ' [] ' markiert werden.

Es ist auch möglich mehrere Befehle einzugeben, indem sie mit der Tasten-  
kombination <Ctrl LCommand J> getrennt werden, was der Return-Taste entspricht  
ohne die Eingabe zu beenden. Das drücken von <LCommand> ist nur dann nötig,  
wenn die Option 'Filter Texteingabefeld' in dem "IControl-Voreinsteller"  
aktiviert wurde. Die Programme werden dann synchron gestartet; man kann aber  
auch "run" benutzen. Bitte beachten, daß die Argumente nur einem Befehl  
übergeben werden. Das hier beschriebene wird intern durch die DOS-Funktion  
System() ausgeführt, deshalb funktioniert es auch mit dem Menüpunkt  
'Befehl ausführen ...' der Workbench oder in der Shell.



## 1.8 Symbol: '>' - Befehl wählen

Zeige ein Dateiauswahlfenster. Hier kann ein neuer Text für Befehl gewählt werden.

## 1.9 Symbol: Arg

Wählen Sie, ob und woher das Befehl seine Argumente erhält. Wenn eins der Textfeldern aktiviert ist, kann die Auswahl auch mit den Tasten <RCommand G> und <RCommand Shift G> erfolgen. Die Möglichkeiten sind:

|                |                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Keine          | Der Befehl wird gleich ausgeführt ohne zusätzliche Argumente.                                                                                                                                          |
| Auswahlfenster | Wenn Sie das Menüpunkt aktivieren, erscheint ein Dateiauswahlfenster, wo Sie die Namen der Datei als Argument für den Befehl bestimmen können. Sie können auch die Ausführung mit 'Abbrechen' stoppen. |
| Piktogramme    | Die Namen der ausgewählten Piktogramme in der Workbench (wenn es welche gibt) werden zu Argumenten.                                                                                                    |
| Pikt/Auswahl   | Wie vorher. Sind keine Piktogramme ausgewählt, dann wie unter 'Auswahlfenster'.                                                                                                                        |

## 1.10 Symbol: Neu

Füge ein neues Menüpunkt in die Liste hinzu. Geben Sie den neuen Befehl an, wählen Sie das Arg Blättersymbol und tippen Sie den Namen des Menüpunkts in den Textfeld unter der Menüpunkte Rolliste ein.

Um ein Programm schnell hinzufügen, ziehen Sie sein Piktogramm ins

Menüpunkte Symbol. Soll das neue Menüpunkt vor einem bestimmten Punkt in die Liste einzugefügt werden, wählen Sie dieses Punkt zuerst.

Die Anzahl der Menüpunkte, die Sie erstellen können, ist nur durch die verfügbare Speicher begrenzt. Beachten Sie aber, daß das jetzige System die Anzahl der anwählbaren Menüpunkte auf 63 begrenzt.

## 1.11 Symbol: Lösch

Lösche das gewählte Menüpunkt aus der Liste.

## 1.12 Symbol: Speichern

Speichere die neuen Menüpunkte in "ENVARC:AppMenu/AppMenu.prefs" ab.

"ENVARC:" liegt normalerweise auf einer Festplatte oder einer Diskette, d. h. die Datei bleibt erhalten. Sie wird beim nächsten System-Start nach "ENV:" (Im RAM-Disk, also im Speicher) kopiert und von AppMenu benutzt. Weiter wie mit

Benutzen

.

## 1.13 Symbol: Benutzen

Speichere die neuen Menüpunkte in "ENV:AppMenu/AppMenu.prefs" ab und beende das AppMenu. Die Datei wird dann von AppMenuCx benutzt. Da "ENV:" im RAM-Disk, also im Speicher des Rechner liegt, ist das neue Menü nur diesmal gültig.

## 1.14 Menü: Projekt/beenden; Symbol: Abbrechen

Beende AppMenu. Wenn die Liste der Menüpunkte geändert wurde und nicht gespeichert ist, erscheint ein Dialogfenster zuerst, mit der Wahl: 'Beenden', 'Speichern und Beenden' oder 'Abbrechen'.

## 1.15 Menü: Projekt/öffnen ...

Öffne die gewählte Datei und lade neue Menüpunkte daraus. Falls die alten

nicht gespeichert wurden, erscheint ein Dialogfenster zuerst, mit der Wahl: 'Öffnen', 'Speichern, dann Öffnen' oder 'Abbrechen'. Es kann auch das Piktogramm der Datei ins Fenster (außer ins

Menüpunkte

Symbol) gezogen werden.

## 1.16 Menü: Projekt/speichern als ...

Speichere die Menüpunkte in einer gewählten Datei ab.

## 1.17 Menu: Vorgaben/Ausschneiden

Kopiere den Text des gerade gewählten Menüpunktes in den  $\leftrightarrow$  Clipboard (wie mit

Kopieren  
) und lösche das Punkt aus der Liste.

## 1.18 Menü: Vorgaben/Kopieren

Kopiere den Text des gerade gewählten Menüpunktes in den  $\leftrightarrow$  Clipboard (wie

Text markieren und `<RCommand C>` in der Shell). Es wird alles in der Form: `'<Befehl-Text>;<Argumente-Beschreibung><Menüpunkt-Text>'` kopiert. Somit ist es möglich, das Menüpunkt zu Löschen und es wo anders in die Liste mit

Neu  
und  
Einfügen  
plazieren. Der Befehl kann auch (wegen `';`) direkt in die Shell geholt werden.

## 1.19 Menü: Vorgaben/Einfügen

Ändere den Befehlstext des gerade gewählten Menüpunktes. Der  $\leftrightarrow$  Text wir aus

dem Clipboard geholt (wie mit `<RCommand V>` in der Shell). Falls der Text die, wie bei

Kopieren  
beschriebene, Form hat, wird auch der Rest geändert.

## 1.20 Menü: Vorgaben/Bewegen rauf, Vorgaben/Bewegen runter

Verschiebe das gerade gewählte Menüpunkt in der Liste um eine Stelle höher bzw. tiefer.

## 1.21 Menü: Vorgaben/auf zuletzt gespeichertes

Lade die zuletzt mit  
Speichern  
gespeicherten Menüpunkte.

## 1.22 Menü: Vorgaben/auf vorherigen Stand

Mache alle Änderungen in der Liste seit dem start von AppMenu rückgängig.

## 1.23 Menü: Optionen/Piktogramme erzeugen?

Bestimmt ob beim Abspeicher mit Speichern als ... ein Piktogramm für eine Datei erzeugt werden soll.

## 1.24 Fehler-Meldungen

Die Programme können folgend Fehler-Meldungen ausgeben.

AppMenu:

- Befehl zu lang  
Der gewählten Befehl ist zu lang. Die maximale Länge ist vom System auf 512 festgelegt.
- Fehler beim Einfügen aus dem Clip
- Fehler beim Kopieren ins Clip  
Konnte Clipboard nicht öffnen, oder ein Fehler beim lesen bzw. schreiben. Einfügen bzw. Kopieren abgebrochen. Läuft weiter.
- Konnte App-Fenster nicht aktivieren  
Es ist kein Einfügen durch plazieren von Piktogrammen im Fenster möglich. Läuft weiter.
- Konnte Dateiauswahlfenster nicht öffnen  
Die Library: asl.library (V37) konnte nicht geöffnet werden, oder nicht genug Speicher für Dateiauswahlfenster. Die gewählte Funktion wird abgebrochen, das Programm läuft weiter.
- Konnte Fenster nicht öffnen  
Zu wenig Speicher für Fenster. Abbruch.
- Lesefehler der Datei <Name>  
Probleme beim lesen der Datei oder zu wenig Speicher für neue Vorgaben.~Läuft weiter.
- Speicherplatzmangel  
Zu wenig Speicher für das Programm oder eine gewählte Funktion. Im ersten Fall Abbruch des Programms, im zweiten der Funktion.
- Schreibfehler der Datei <Name>  
Probleme beim speichern der Datei. Läuft weiter.
- Ungültige residente Library  
Die Version für WB 3.x wurde auf 2.x gestartet oder konnte eine der Libraries nicht öffnen: intuition.library (V37), gadtools.library (V37), workbench.library (V37), icon.library (V37), iffparse.library (V37). Abbruch.

AppMenuCx:

- Befehl zu lang  
Die gewählten Argumente machen den Befehl zu lang. Die maximale Länge ist vom System auf 512 festgelegt. Der Befehl wird nicht ausgeführt.
  - Konnte alle Menüpunkte nicht erzeugen  
Zu wenig Speicher um alle Menüpunkte zu erzeugen. Läuft weiter.
  - Konnte Cx-Objekt nicht erzeugen  
Konnte das Programm als Commodity nicht anmelden. Abbruch.
  - Konnte Dateiauswahlfenster nicht öffnen  
Die Library: asl.library (V37) konnte nicht geöffnet werden, oder nicht genug Speicher für Dateiauswahlfenster. Die gewählte Funktion wird abgebrochen, das Programm läuft weiter.
-

- Konnte Notifikation nicht setzen für Datei "ENV:AppMenu/AppMenu.prefs"  
Zu wenig Speicher oder Probleme mit der Datei. Abbruch.
- Lesefehler der Datei "ENV:AppMenu/AppMenu.prefs"  
Probleme beim Lesen der Datei oder zu wenig Speicher für neue  
Vorgaben.~Abbruch.
- Speicherplatzmangel  
Zu wenig Speicher für das Programm oder eine gewählte Funktion. Im  
ersten Fall Abbruch des Programms, im zweiten der Funktion.
- Ungültige residente Library  
Die Version für WB 3.x wurde auf 2.x gestartet oder konnte eine der  
Libraries nicht öffnen: intuition.library (V37), commodities.library  
(V37), workbench.library (V37), icon.library (V37), iffparse.library  
(V37). Abbruch.

## 1.25 Autor und Programme

Falls Sie Kommentare, Ideen, Fehler-Meldungen zu den Programmen haben, oder meine Arbeit belohnen wollen, dann schreiben Sie mir oder rufen Sie mich an.

Ich bin erreichbar per

Post: Marius Chwalczyk  
Ziegelstr. 59/165  
D-67655 Kaiserslautern  
Deutschland

Tel.: 0631-10944

AppMenu und AppMenuCx sind in C geschrieben und wurden mit SAS/C 6.55 kompiliert. Sie wurden auf einem Amiga 3000 (2.0 und 3.1) mit "Enforcer" und "Mungwall" getestet.

Wenn Sie Katalogdateien Ihrer Sprache erstellen wollen, benutzen Sie die beigelegten Katalogbeschreibungen. Bitte darauf achten, daß die Texte mit standard Font "Topaz/8" in alle Fenstersymbole passen. Ich bin bereit Ihre Übersetzung der nächsten Freigabe beizufügen.

Die deutsche Katalog-Datei wurde mit Hilfe des Programms "KitCat 1.1" von Albert Weinert erstellt.

## 1.26 Geschichte and was neues

NEU = neues hinzugefügt, ÄND = etwas geändert, KOR = Fehler korrigiert

2.4 (8.10.95)

- KOR: "Arg:Pikt/AuswahlF" funktionierte nicht

2.3 (3.4.95)

- NEU: zwei Versionen der Programmen, für WB 2.x und 3.x
- ÄND: benutze Memory Pool funktionen: V37 amiga.lib, V39 exec
- ÄND: deutsche Kataloge

## AppMenu:

- ÄND: AppWindow Verhalten geändert: um neues Menüpunkt hinzufügen ziehe Piktogramm ins Listen-Symbol, um Vorgaben zu laden ziehe ihr Piktogramm woanders
- ÄND: Größe der Symbole wird ganz dem Zeichensatz angepaßt
- ÄND: Menü-Kurzbeefehle und -Texte dem OS angepaßt
- ÄND: Vorgabe von "Piktogramm erzeugen?" ist YES bei WB-Start und NO bei Shell-Start
- NEU: Piktogramm-Dateien werden im Dateiauswahlfenster nicht mehr angezeigt
- NEU: "Vorgaben/Einfügen" Menüpunkt und CLIPUNIT/K/N Argument
- NEU: CTRL-F Taste um das Fenster zu "wecken"
- KOR: NewLookMenu Problem (Tag war in CreateMenus()), jetzt korrekt in LayoutMenus())

## 2.2 (31.8.94)

- nichts neues, außer ...

Tja, hier ein paar technische Erklärungen: Ich bin jetzt ziemlich sicher, daß beide Programme der Version 2.1 die Rechner mit dem 68000-Prozessor zum Absturz bringen können. Ich kann es leider nicht testen, da ich einen '030 in meinem A3000 habe. Es kann nämlich einen 80000003 Guru ("Illegal address access (ie: odd)", dt. "Illegaler zugriff auf Adresse (z.B. ungerade)") geben. Die Programme können in Wort (UWORD) von einer ungeraden Adresse lesen. Auf meinem Rechner geht es glatt, aber auf einem 68000 ist es verboten!

Als ich die Lokalisierung machte, benutzte ich die ersten zwei Bytes (ein Wort) der Strings als Index-Nummer. So ungefähr: "ID = \*(UWORD\*)STR\_ID; STR = STR\_ID + 2;". Ich habe jetzt festgestellt, daß der Compiler (SAS/C 6.51) die als "static const UBYTE str[];" deklarierten Strings auch auf ungeraden Adressen aufstellt. Für einen String (Byte-Array) ist es OK, da man ein Byte auch von solchen Adressen lesen darf. Bei einem Wort gibt es wie gesagt Probleme. Jetzt müßte alles funktionieren, da ich nur die ersten Bytes für die Nummern benutze.

Die Idee mit dem Wort-Index hatte ich übrigens aus dem Programm "Cat2H" von Niko François. Sie war zwar gut, aber nicht ganz :-). Also wer das Programm benutzt, soll hiermit gewarnt werden!

## 2.1 (18.7.94)

- NEU: beide Programme lokalisiert
- NEU: deutsche Katalogdateien beigelegt
- ÄND: Änderungen in den Texten und der Größe des Fensters und der Symbole

## 2.0 (14.5.94)

- Programm geteilt in AppMenuCx (Hilfsmittel-Menü) und AppMenu (Voreinsteller)

## AppMenuCx:

- NEU: ' [] ' im Befehl markiert, wo Argumente hinzugefügt werden.
- NEU: mehrere Dateien wählbar und Pattern-Matching im Dateiauswahlfenster.

## AppMenu:

- KOR: Symbole (insbesondere 'Neu') und Menü arbeiten jetzt korrekt
- ÄND: beide SAVEICONS und SCREENFONT haben YES als Vorgabe, sind nur noch als Piktogramm-Merkmale da, und auch beim Shell-Start wirksam.
- ÄND: 'Kopieren' kopiert jetzt alles; 'Einfügen' erstellt kein neues Punkt, sondern ändert das gerade gewählte ('Neu' zuerst benutzen).
- NEU: mehrfache Auswahl im Menü und mehr Tastatur-Kurzbeefehle
- NEU: Tastatur-Kurzbeefehle sind bei aktivierten Textfeldern wirksam
- NEU: wenn eins der Textfeldern aktiviert ist, <RCommand G> für nächsten

und <RCommand Shift G> für vorherige 'Arg'.

- NEU: Symbol für Dateiauswahlfenster um 'Befehl' zu ändern.

1.x (25.3.93 - 4.7.93)

[alter Text gelöscht]

---