

AppMenu_D

COLLABORATORS

	<i>TITLE :</i> AppMenu_D		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AppMenu_D	1
1.1	AppMenu.dok	1
1.2	Copyright	1
1.3	Einführung	1
1.4	Installieren und starten	2
1.5	Benutzen	2
1.6	Menu Items	3
1.7	Command	3
1.8	Arg	3
1.9	New	4
1.10	Del	4
1.11	Save	4
1.12	Use	4
1.13	Cancel	4
1.14	Quit	5
1.15	Open...	5
1.16	Save As...	5
1.17	Last Saved	5
1.18	Restore	5
1.19	Copy To Clip	5
1.20	Paste From Clip	5
1.21	Move Item Up/Down	5
1.22	Das Programm und sein Autor	6
1.23	Geschichte	6

Chapter 1

AppMenu_D

1.1 AppMenu.dok

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

\$VER: AppMenu 1.4 (4.7.93)

COPYRIGHT
EINFÜHRUNG
INSTALLIEREN + STARTEN
BENUTZEN
PROGRAMM + AUTOR
GESCHICHTE

WICHTIG: Das Format der Voreinstellungsdatei wurde von ASCII (in den Versionen 1.1 und 1.2) in IFF PREF geändert. Bitte das Programm "ConvertPrefs" benutzen, um diese alten Versionen der Dateien zu konvertieren.

1.2 Copyright

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

AppMenu ist copyrighted © 1993 by Marius Chwalczyk. Es kann frei weitergegeben werden solange alle Dateien in ihrem originellen Zustand ohne irgendwelche Änderungen enthalten sind. Nur minimale Kopierkosten (Datenträgerpreis) sind bei der Weitergabe erlaubt. Das Programm ist "wie es ist", jede Benutzung ist auf eigene Risiko.

1.3 Einführung

AppMenu ist ein Workbench Utility. Es läuft nur unter Release 2 (V37) und späteren Versionen von AmigaOS. Es ist ein Commodity mit dem eigene Menüpunkte in das Tools Menü von Workbench eintragen werden können. Wird ein Menüpunkt angewählt, ruft AppMenu das zugehörige Programm auf. Alle Menüpunkte und die Befehle können frei definiert werden. Alle Programme werden als eigene,

asynchrone Prozesse gestartet. Es können Argumente für diese Programme durch Piktogramme oder Dateiauswahlfenster angegeben werden.

1.4 Installieren und starten

Um AppMenu zu installieren, kopieren Sie das Programm in das Verzeichnis ihrer Wahl. Starten können Sie es über Workbench oder über Shell. Sie können es auch in dem WBStartup Ordner speichern. AppMenu versteht folgende Argumente bei Shell oder Merkmale des Piktogramms:

```
CX_PRIORITY=<n>
CX_POPUP=<NO|YES>
CX_POPKEY=<Tastebefehl>
```

Siehe im Workbench Handbuch unter Comodities. Die Vorgaben sind Priorität 0, Popup 'NO' und Tastebefehl 'shift f3'

```
FROM=<Dateiname>
```

'<Dateiname>' bezeichnet die Voreinstellungsdatei (vorgegeben ist "ENVARC:AppMenu/AppMenu.prefs"), von wo AppMenu die Beschreibung der Menüpunkte liest und wo sie gespeichert werden. Jeder Menüpunkt enthält einen Text für das Menü, einen für den Befehl und eine Spezifizierung der Argumente.

```
PROGITEM=<NO|YES>
```

Wenn kein "AppMenu..."-punkt im Tools Menü erwünscht, bitte NO wählen. Die Vorgabe ist YES.

```
SCREENFONT=<NO|YES>
```

Wollen Sie, daß AppMenu den voreingestellten Fenster-Zeichensatz statt den "Topaz/8" benutzt, dann geben Sie YES an. Bitte beachten, daß nur die Höhe der Symbole dem Zeichensatz angapapt wird. Wenn er also zu breit ist, können Texte über die Symbole hinweggeschrieben werden. Vorgegeben ist NO.

1.5 Benutzen

Das Fenster von AppMenu enthält eine Rollliste mit ihrem Rollbalken und einem Textfeld darunter, ein weiteres Textfeld, ein Blättersymbol, zwei Aktionssymbole und eine Reihe von vier Aktionssymbolen. Mit der rechten Maustaste ist auch ein Menü mit weiteren Aktionen erreichbar.

Um das Programmfenster zu öffnen, wählen Sie den Tastebefehl, aktivieren Sie das "AppMenu..."-Punkt in dem Tools-Menü, rufen Sie AppMenu nochmal auf oder benutzen Sie das "Exchange" Programm. "Exchange" können Sie auch für anderen Funktionen benutzen (siehe seine Dokumentation).

Es gibt folgende Symbole:

New	Del	Menu Items
		Arg
	Command	

Save Use Cancel Quit

Und folgende Menüpunkte:

Project	Edit
Open...	Last Saved
Save	Restore
Save As...	~~~~~
Use	Copy To Clip
About...	Paste From Clip
~~~~~	~~~~~
Cancel	Move Item Up
Quit	Move Item Down

## 1.6 Menu Items

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 in Command 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 für das Menüpunkt ändern. Nach Betätigung der Eingabetaste wird das Menüpunkt in die Liste zurückgenommen.

## 1.7 Command

Geben Sie hier den Namen des Programms und seine Argumente ein, das Sie starten wollen. Sie müssen den ganzen Pfad des Befehls angeben, wenn AppMenu über Workbench gestartet wird.

Es ist auch möglich mehrere Programme einzugeben, indem sie mit der Tastenkombination <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 'Text gadget filter' in der "IControl" Preferences-Datei aktiviert wurde. Die Programme werden dann synchron gestartet; man kann aber auch "run" benutzen. Bitte beachten, daß die Argumente nur dem letzten Programm übergeben werden. Das hier beschriebene wird intern durch die DOS-Funktion System() ausgeführt, deshalb funktioniert es auch mit dem Menüpunkt 'Execute Command...' der Workbench oder in der Shell.

## 1.8 Arg

Wählen Sie, ob und woher Sie Argumente für ein Befehl brauchen. Die Möglichkeiten sind:

Requester    Wenn Sie das Menüpunkt aktivieren, erscheint ein Dateiauswahlfenster, wo Sie den Namen einer Datei als Argument für den Befehl bestimmen können. Sie können auch die Ausführung mit 'Cancel' abbrechen.

Icons	Die Namen der ausgewählten Piktogramme in der Workbench (wenn es welche gibt) werden zu Argumenten.
Icons/Req	Wie vorher. Sind keine Piktogramme ausgewählt, dann wie unter 'Requester'.
None	Der Befehl wird gleich ausgeführt ohne zusätzliche Argumente.

## 1.9 New

Füge ein neues Menüpunkt in die Liste hinzu. Geben Sie den Text für einen neuen Befehl in Command an, wählen Sie das Arg Blättersymbol und tippen Sie den Namen des Menüpunkts in den Textfeld unter der {" Menu Items " link AM-3.0} ← Rollliste ein. Um

Programme schnell hinzuzufügen, ziehen Sie ihre Piktogramme in das Fenster. Soll ein neues Menüpunkt vor einem bestimmten Punkt in die Liste einzuggefügt 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 System (V37) die Anzahl der anwählbaren Menüpunkte auf 63 begrenzt.

## 1.10 Del

Lösche das gewählte Menüpunkt aus der Liste.

## 1.11 Save

Speichere die Menüpunkte ab. Weiter wie Use .

## 1.12 Use

Schließe das Fenster, aktiviere die Menüpunkte in dem Tools-Menü und führe die Befähle der gewählten Punkte aus.

## 1.13 Cancel

Mache alle Änderungen seit dem Öffnen des Fensters rückgängig, schließe es und benutze die alte Liste. Das Schließ-Symbol und Hide-Symbol von "Exchange" haben die gleiche Funktion.

---

## 1.14 Quit

Beende AppMenu. Wenn die geänderten Menüpunkte nicht gespeichert sind, erscheint ein Dialogfenster zuerst.

## 1.15 Open...

Lade neue Menüpunkte aus einer gewählten Datei. Falls die alten nicht gespeichert wurden, erscheint ein Dialogfenster zuerst.

## 1.16 Save As...

Speichere die Menüpunkte in einer gewählten Datei ab.

## 1.17 Last Saved

Lade die zuletzt gespeicherten Menüpunkte.

## 1.18 Restore

Mache alle Änderungen seit dem Öffnen des Fensters rückgängig.

## 1.19 Copy To Clip

Übertrage den Befehlstext in ein Kopierpuffer (Clip). Er kann dann z.B. in die Shell mit `<RCommand V>` geholt werden.

## 1.20 Paste From Clip

Füge ein neues Menüpunkt in die Liste hinzu. Sein Befehlstext wird von dem Kopierpuffer (Clip) geholt.

## 1.21 Move Item Up/Down

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

---

## 1.22 Das Programm und sein Autor

Ja, ja..., die Lokalisierung sollte gemacht werden. Der 2.1-Update ist aber immer noch nicht (in Deutschland) erhältlich. Jetzt muß ich wohl auf WB 3.1 warten, aber auf so was hat uns Commodore leider schon gewöhnt. :-)

AppMenu ist in C geschrieben und wurde mit SAS/C 6.3 kompiliert. Es wurde auf einem Amiga 3000 (KS 37.175, WB 37.71) mit "Enforcer" und "Mungwall" getestet.

Falls Sie Kommentare, Ideen und/oder Bug-Meldungen haben, dann schreiben Sie mir oder rufen Sie mich an.

Ich bin erreichbar per

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

Tel.: 0631-10944

## 1.23 Geschichte

1.0 erste interne Version.

1.1 (25.3.93) - erste Public-Version.

- Speichert Voreinstellungen in separater Datei und nicht mit dem Piktogramm.
- Voreinstellungen-nicht-gespeichert-Dialogfenster hinzugefügt.
- 'Icons' als Argumente hinzugefügt.

1.2 (29.4.93)

- Keine Bugs gefunden oder gemeldet - keine Bugs korrigiert! :-)
- Möglichkeit zum Ordnen von Punkten in der Liste.  
(Danke an Kevin A. Kirk für den Vorschlag.)
- Neues Programmargument: PROGITEM.
- Dateiauswahlfenster merkt sich frühere Einstellungen.

1.3 (2.6.93)

- Format der Voreinstellungsdatei in IFF PREF geändert, "ConvertPrefs" zum konvertieren der alten Dateien geschrieben.
- Die Voreinstellungen werden jetzt zeitweilig in "ENV:AppMenu/AppMenu.prefs" gehalten.
- Neue Funktionen: 'Open...', 'Save As...', 'Last Saved', 'Restore', 'Copy To Clip', 'Paste From Clip', 'Move Item Up/Down'
- Bug korr.: Voreinstellungen waren immer in "ENVARC:AppMenu/AppMenu.prefs" gespeichert.
- Bug korr.: Enable/Disable von "Exchange" funktionierte falsch (beide Bugs nur in 1.2, wegen einer "ungetesteten Code-Optimierung").
- Code optimiert aber diesmal getestet. :-)

1.3b (15.6.93)

---

- Wenn keine Voreinstellungsdatei existierte, AppMenu 1.3 meldete 'Error while loading prefs' und brach ab. Jetzt läuft es weiter nach der Meldung.

#### 1.4 (4.7.93)

- Bug korr.: wenn mehrere Piktogramme zusammen in das Fenster gezogen wurden und manche Leerzeichen in ihren Namen hatten, ihren Befehle wurden falsch in den Ausrufungszeichen geschrieben.
  - Nach 'Paste From Clip' oder 'New' die Menüs 'Move Item Up/Down' werden jetzt immer in "Geisterschrift" angezeigt.
  - Parsing optimiert (benutzt ReadArgs()/FindToolType() statt ArgXXX() jetzt).
  - Neues Programmargument: SCREENFONT.
-