

**changeicon**

**COLLABORATORS**

	<i>TITLE :</i> changeicon		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		September 19, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>changeicon</b>	<b>1</b>
1.1	Inhaltsverzeichnis . . . . .	1
1.2	Einleitung . . . . .	2
1.3	Copyright . . . . .	2
1.4	Voraussetzungen . . . . .	3
1.5	Installation . . . . .	4
1.6	Programmstart . . . . .	4
1.7	Bedienung . . . . .	6
1.8	Der Rest . . . . .	9
1.9	Adresse des Autors . . . . .	9
1.10	Workbench Applikationen . . . . .	9
1.11	ChangeIcon übersetzen . . . . .	10
1.12	ToolTypes . . . . .	10
1.13	Eingebaute Typen . . . . .	11
1.14	Index . . . . .	11

---

# Chapter 1

## changeicon

### 1.1 Inhaltsverzeichnis

```

                                     ### #
# #                                     #
# ##### ##### # ### ##### # ### # ### # ### # ### #
# # # # # # # # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # # # # # # #
# # # # # # # # # # # # # # # # # # # # #
### # # ### # # # ### ##### ### ### ### ### # # V1.0
#
###

```

Inhalt:

```

Einleitung
    Worum geht's hier überhaupt??

Copyright
    DIESEN TEXT UNBEDINGT LESEN!

Voraussetzungen
    Was benötigt ChangeIcon?

Installation
    Wie mach ich das Ding lauffähig?

Programmstart
    Wie läuft's denn nun?

Bedienung
    Wie arbeite ich mit ChangeIcon?

Der Rest
    Danksagung, History, Zukunft

Meine Adresse
    Wer hat das alles denn verzapft?

```

Index

## 1.2 Einleitung

ChangeIcon ist ein Programm, welches sich als  
 » Workbench-Applikation  
 installiert. Der User kann nun Piktogramme auf ChangeIcon ↔  
 werfen, welche

dann je nach dem Typ der Datei durch andere Piktogramme ersetzt werden.  
 Sinnvoller wird das ganze, wenn Dateien noch gar keine Piktogramme besitzen.  
 Auch diesen können Piktogramme zugewiesen werden, wenn man auf der Workbench  
 den Menüpunkt "Inhalt anzeigen -> alle Dateien" im "Fenster"-Menü anwählt.

Sicher gibt es schon mehrere ähnliche Programme, jedoch konnte mich keines  
 überzeugen. Zudem wollte ich mich mit der Programmierung der locale.library  
 und der whatis.library vertraut machen. Letztere dient der Erkennung von  
 Datei-Typen und läßt den Benutzer auf einfache Weise eigene Typen kreieren.

Du wirst sehen, beim Erstellen von Disketten, die zum Beispiel an Freunde  
 verschickt werden, erspart man sich eine Menge Arbeit, wenn man die Diskette  
 zunächst ohne Piktogramme erstellt und dann ChangeIcon werkeln läßt!

## 1.3 Copyright

Commodore ist ein eingetragenes Warenzeichen von Commodore ↔  
 Electronics Ltd.

Amiga und Amiga Workbench sind eingetragene Warenzeichen von  
 Commodore-Amiga, Inc. AmigaGuide und Installer stehen unter dem (C)opyright  
 von Commodore-Amiga, Inc.

ChangeIcon ist Freeware. Das Copyright liegt bei mir, Hanns Holger Rutz. Für  
 die Verbreitung von ChangeIcon müssen mehrere Regeln beachtet werden, deren  
 Nichteinhaltung verboten ist:

- ChangeIcon darf nur als ganzes verbreitet werden. Folgende Dateien müssen  
 unverändert in der dargestellten Verzeichnisstruktur weitergegeben werden:

```
ChangeIcon (dir)
  Catalogs (dir)
    deutsch (dir)
      changeicon.catalog           ; Sprach-Datei deutsch
    français (dir)
      changeicon.catalog           ; Sprach-Datei französisch
  ChangeIcon.cd                   ; CatalogDescription-Datei
  ChangeIcon.ct                   ; CatalogTranslation-Datei
  Docs (dir)
    Deutsch                       ; die deutsche Anleitung
    Deutsch.info                   ; zugehöriges Piktogramm
    English                       ; die englische Anleitung
    English.info                   ; zugehöriges Piktogramm
    Français                      ; die französische Anleitung
    Français.info                 ; zugehöriges Piktogramm
```

```

*Icons (dir)                ; enthält Beispiel-Piktogramme
  Install (dir)
    Deutsch.info            ; deutsches Installer-Piktogramm
    English.info            ; englisches Installer-Piktogramm
    Français.info           ; französisches Piktogramm
    Install.script          ; Installer-Script
  S (dir)
    FileTypes               ; Beispiel-Datei für Dateitypen
  ChangeIcon                 ; das Programm selbst
  ChangeIcon.info           ; Piktogramm für ChangeIcon
*ChangeIcon_Source.lha     ; der Quellcode, gepackt
  Docs.info                 ; (beliebiges) Ordner-Piktogramm
  Install.info              ; (beliebiges) Ordner-Piktogramm
  ReadMe                    ; FD-Serien abhängige Zusatzinfos
  ReadMe.info               ; zugehöriges Piktogramm

```

Mit einem Sternchen versehene Dateien und Verzeichnisse können optional mitkopiert werden. Das Packen dieses gesamten Verzeichnisses (beispielsweise für DFÜ-Zwecke) ist natürlich erlaubt.

- das ChangeIcon-Paket darf nur auf FD-Serien vertrieben werden, wenn der Ersteller der Serie eine schriftliche Lizenz bei mir eingeholt hat. Der kommerzielle Verkauf des Pakets ist verboten. Dies ist leider nötig, da es immer noch genügend Firmen gibt, die FD-Programme für teures Geld verkaufen – der Autor sieht natürlich keine Mark. Fred Fish darf das Paket auf seinen AmigaLibDisks bzw. seinen CD-ROMs verbreiten.
- das ChangeIcon-Paket darf via DFÜ verbreitet werden, eine Lizenz ist nicht notwendig.
- ein evtl. vorhandener Quellcode darf nur zu Lernzwecken benutzt werden. Neu assemblierte Programme oder geänderte Quellcode-Dateien dürfen nicht weitergegeben werden.

WICHTIG: DER AUTOR, HANNS HOLGER RUTZ, HAFTET FÜR KEINERLEI DURCH DEN GEBRAUCH VON CHANGEICON ENTSTANDENE SCHÄDEN. DAS TUT ALLEINE DER BENUTZER!

Natürlich bin ich dankbar für Bugreports und Vorschläge

```

(
    » Adresse des Autors
)!

```

## 1.4 Voraussetzungen

ChangeIcon benötigt mindestens OS2.04. Um ChangeIcon in einer anderen Spra-

che als englisch zu benutzen, wird die locale.library V38 benötigt, welche sich ab Version 2.1 auf der Workbench befindet.

In diesem Zusammenhang möchte ich Dich bitten, die CatalogTranslation-Datei ("ChangeIcon.ct") in eine Sprache zu übersetzen, derer Du mächtig bist und zu der noch keine Sprach-Datei existiert. Lies dazu

```

» Übersetzung erstellen
!
```

Ferner baut ChangeIcon auf der `whatis.library` (Version 2 und höher) auf. Das Copyright des WhatIs-Pakets liegt bei Sylvain Rougier und Pierre Carette. Beachte, daß, falls das WhatIs-Paket mitgeliefert wird, dafür die Vertriebs-Konditionen von WhatIs gelten!

Der Geschwindigkeit halber empfehle ich außerdem eine Festplatte.

## 1.5 Installation

Die Installation wird im Normalfall mit Hilfe eines der Installer-Scripts im "Install" Verzeichnis durchgeführt. Dazu muß sich der Commodore-Installer in einem Verzeichnis befinden, das als Programm-Pfad definiert wurde, zum Beispiel "C:" oder "SYS:Utilities/".

Will man die Installation per Hand durchführen, ist folgendes zu beachten:

- die `whatis.library` muß, falls sie beiliegt, ins "LIBS:" Verzeichnis kopiert werden.
- die Beispiel-Datei für Dateitypen, "S/FileTypes", muß nach "S:" kopiert werden, falls man eine solche Datei noch nicht hat.
- für die deutsche Programmführung muß die Datei "Catalogs/deutsch/changeicon.catalog" nach "LOCALE:Catalogs/deutsch/" kopiert sowie deutsch als Sprache im Voreinsteller "Locale" gewählt werden.

## 1.6 Programmstart

ChangeIcon muß von der Workbench gestartet werden, für die Programmierung des Shell-Starts war ich zu faul :-). Folgende » ToolTypes werden unterstützt:

MODE=ICON|MENU entscheidet, wie ChangeIcon sich verhält, wenn sein Fenster geschlossen ist. Bei ICON wird ein AppIcon benutzt, bei MENU wird ein AppMenuItem benutzt. Voreingestellt: ICON.  
(  
» Workbench Applikationen  
)

POPUP=YES|NO soll ChangeIcon beim Start das Fenster mit weiteren Optionen öffnen oder im AppIcon bzw. AppMenuItem Modus erscheinen? Bei YES wird das Fenster geöffnet. Voreingestellt: NO.

EDITOR=<Pfad> ChangeIcon bietet die Möglichkeit, die Datei, welche die Dateitypen beschreibt, zu editie-

ren. <Pfad> gibt den Namen des Editors an, der dafür benutzt werden soll, zum Beispiel "SYS:Tools/MEmacs". Wenn dieser ToolType nicht angegeben wird, richtet sich ChangeIcon nach der Environment-Variablen "EDITOR". Wenn diese nicht existiert, wird "C:Ed" als Editor benutzt.

ICONDIR=<Verzeichnis> dieser ToolType ist sehr wichtig. Mit <Verzeichnis> gibt man an, wo die Piktogramme liegen, die die verschiedenen Dateitypen repräsentieren. Beispiel: "Work:Icons/". Mehr dazu in

» Bedienung

ICONNAME=<Pfad>|<Name> <Name> gibt den Text an, welcher unter dem AppIcon von ChangeIcon erscheint. Will man ein anderes Piktogramm als das von ChangeIcon für das AppIcon verwenden, kann man mit <Pfad> dessen Namen angeben, der Dateiname erscheint dann unter dem AppIcon. Beispiel dafür: "Work:Icons/ChangeIcon\_AppIcon". Voreingestellt ist "ChangeIcon". Dieser ToolType wird ignoriert, wenn MODE=MENU eingestellt wurde.

ICON\_X=<Wert> Mit ICON\_X und  
ICON\_Y=<Wert> ICON\_Y kann man die Position des AppIcons im Workbench-Fenster angeben. Die Werte müssen in Pixels (ICON\_X) bzw. in Rasterzeilen (ICON\_Y) angegeben werden. Wenn diese ToolTypes nicht benutzt werden, erscheint das AppIcon auf einem freien Platz im Workbench-Fenster. Dieser ToolType wird ignoriert, wenn MODE=MENU eingestellt wurde.

MENUNAME=<Name> Dieser ToolType wird nur beachtet, wenn MODE=MENU eingestellt wurde. Dann gibt <Name> den Text an, der im "Hilfsmittel" Menü der Workbench erscheinen soll. Voreingestellt ist "ChangeIcon".

DONOTWAIT Dieser ToolType muß angegeben werden, wenn Du ChangeIcon in den "WBStartup" Ordner Deiner Workbench legst, damit es automatisch beim Booten gestartet wird.

Wenn Du Deine gewünschten Parameter eingestellt hast, kannst Du ChangeIcon einfach durch Doppelklick auf sein Piktogramm starten. Je nach gewähltem MODE- und POPUP-ToolType erscheint dann das Fenster, das AppIcon oder das AppMenuItem von ChangeIcon. Wenn ein Fehler beim Start auftrat, erscheint eine Meldung und man hat evtl. die Möglichkeit, es noch einmal zu versuchen.

(

» Bedienung

)

## 1.7 Bedienung

Der Einfachheit halber gehe ich jetzt einmal davon aus, daß Du ←  
den ToolType

» POPUP=YES

eingestellt hast. Dann erscheint bei erfolgreichem Start das Fenster von ChangeIcon unter dem Mauszeiger. Folgende Gadgets (Schalter) bietet die Oberfläche (unterstrichene Zeichen bedeuten, daß der Schalter auch durch Druck auf die entsprechende Taste bedient werden kann):

**Auto Typ** Wenn der Schalter abgehakt ist, entscheidet ChangeIcon (bzw. die `whatis.library`) selbst, um welche Arten von Dateien es sich bei den Piktogrammen handelt, die der Benutzer in das Fenster zieht. Bei den ermittelten Typen handelt es sich um die, die in der Liste am linken Fensterrand aufgeführt sind.

**Typen** in dieser Liste sind alle Typen aufgeführt, die momentan zur Verfügung stehen. Wählt man einen per Mausklick aus, wird "Auto Typ" ausgeschaltet. Wirft der Benutzer jetzt Piktogramme in das Fenster, wird das Piktogramm für den in der Liste ausgewählten Typen vergeben.  
Die Typen können auch mit den Cursor-Tasten ausgewählt werden, mit Umschalt+Cursor wird eine ganze Seite geblättert und mit Alt+Cursor zum Anfang bzw. Ende.

**Standardprogramm** Bei "Auto Typ" ist dieser Schalter nicht wählbar. Hat man jedoch einen Typen aus der Liste ausgewählt, kann der Schalter abgehakt werden. In diesem Fall wird der Inhalt des String-Gadgets (rechts daneben) als Standardprogramm für das Piktogramm des ausgewählten Typen benutzt. Hat man beispielsweise "Text" als Typen ausgewählt, in dem Text-Piktogramm "def\_Text.info" (dazu gleich mehr) ":C/Look" als Standardprogramm gewählt und will jetzt Piktogramme für Dateien auf einer Diskette erzeugen, die an einen Menschen verschickt werden sollen - nennen wir in Peter -, der nur "MuchMore" benutzt. Also schaltet man die Standardprogramm-Option ein und schreibt in das String-Gadget den Text ":C/MuchMore".  
Das String-Gadget kann mit der Leertaste aktiviert werden.

**AppIcon bzw. AppMenuItem** Hierüber wird das Fenster von ChangeIcon geschlossen. Je nach gewähltem

» MODE

ToolType erscheint jetzt ein Piktogramm im Workbench-Fenster oder ein Menüpunkt im "Hilfsmittel"-Menü der Workbench. Außer der Tatsache, daß dann natürlich keine Optionen verändert werden können, verhält sich ChangeIcon wie bisher: Wenn Du

Piktogramme auf das AppIcon wirfst oder Piktogramme anklickst und dann den Menüpunkt wählst, ändert das Programm die Piktogramme um - war "Auto Typ" angeschaltet, wird der Dateityp ermittelt, ansonsten wird der in der Liste ausgewählte Typ benutzt. Woher ChangeIcon das neue Piktogramm nimmt etc. erfährst Du gleich.

Durch Doppelklick auf das AppIcon oder einfache Wahl des AppMenuItems gelangst Du wieder zum Fenster.

Verlassen Hierüber kann ChangeIcon beendet werden (keine Sicherheitsabfrage).

So. Was passiert denn nun eigentlich, wenn man Piktogramme in das Fenster oder auf das AppIcon wirft? Dazu muß man wissen, daß ChangeIcon die whatis.library benutzt. Diese wiederum greift auf eine Datei Namens "S:FileTypes" zu. Dort sind alle Typen, die das Programm kennen soll, beschrieben (die wichtigsten sind in der Library eingebaut). Um die Typen zu ändern oder neue zu erstellen, startet man einen Texteditor mit der Datei "S:FileTypes". Eine einfache Methode zeige ich gleich bei der Besprechung der Menüs noch.

Ich bespreche den Aufbau der Datei anhand des Beispiels AmigaGuide-Typ. Für genauere Informationen solltest Du die Anleitung der whatis.library lesen.

```
# AmigaGuide-Dateien
TYPE "Guide"
  SUBTYPE      Text
  ICONNAME     "def_Guide"
  OPTNAMEPATTERN "#?.guide"
  SEARCHPATTERN "@DATABASE"
ENDTYPE
```

Die erste Zeile beginnt mit einem Kommentar. Solche werden durch das Nummern-Zeichen '#' eingeleitet. Mit "TYPE" beginnt die Definition eines neuen Dateitypen, hier ist das "Guide". "SUBTYPE" gibt den Namen eines bereits bestehenden Typen an, auf dem der neue basiert. Nicht jeder Typ muß ein Sub-Typ eines anderen sein. "ICONNAME" ist für ChangeIcon sehr wichtig. Hier steht der Name der Piktogramm-Datei, die den Typen repräsentiert. Das Verzeichnis, in dem diese Piktogramme liegen, kann mit dem ToolType

» ICONDIR

angegeben werden. Das "def\_Guide" Piktogramm kann zum Beispiel ←  
mit dem Pro-

gramm IconEdit der Workbench erstellt werden. Die Namen der eingebauten Typen sind immer "def\_<TypName>". Für <TypName> kommen die in

» Eingebaute Typen

aufgeführten Typen in Frage. "OPTNAMEPATTERN" ist für ChangeIcon unwichtig, manche anderen Programme, die die whatis.library benutzen, brauchen diesen Eintrag evtl. Der letzte Befehl schließlich lautet "SEARCHPATTERN". Dies ist wichtig, wenn "Auto Typ" angeschaltet ist. Dann muß ja versucht werden, die Typen der Dateien selbständig zu erkennen. Die AmigaGuide-Dateien erkennt man also daran, daß am Anfang das Wort "@DATABASE" steht (Groß-/Kleinschreibung ist hier egal). Man hat noch weitere Möglichkeiten, die Charakteristika eines Typs festzulegen, ich verweise hier aber auf die WhatIs-Anleitung.

Zurück zu ChangeIcon. Wenn Du jetzt also ein oder mehrere Piktogramme auf

das AppIcon oder in das Fenster wirfst, ermittelt ChangeIcon den Typen der Datei(en) oder nimmt den vom Benutzer festgelegten. Dann wird versucht, das Piktogramm zu laden, dessen Name bei der Typendefinition mit "ICONNAME" spezifiziert wurde. Und zwar sucht ChangeIcon in folgenden Verzeichnissen:

- dem mit dem ToolType
  - » ICONDIR
  - angegebenen
- dem aktuellen
- in "ENV:Sys/"
- in "ENVARC:Sys/"

Sollte der Typ nicht erkannt werden oder das neue Piktogramm nicht gefunden werden, blinkt der Bildschirm auf. Ansonsten werden den Dateien, die in das Fenster bzw. auf das AppIcon geworfen wurden, neue Piktogramme verliehen. Man kann auch Piktogramme für Dateien erzeugen, die noch keine besitzen. Dazu wählt man im gewünschten Ordner den Punkt "Inhalt anzeigen -> alle Dateien" aus dem "Fenster"-Menü der Workbench. Ein weiterer Trick besteht in der Möglichkeit, Typen in der Liste zu führen, die durch "Auto Typ" gar nicht ermittelt werden können - wenn man zum Beispiel zwei Piktogramme für Texte haben will. Der Aufbau der Typendefinition in "S:FileTypes" sollte dann etwa so aussehen:

```
TYPE "Text2"
  ICONNAME      "def_Text2"
  NAMEPATTERN  "~(#?)"      # keine Datei kann so einen Namen haben :-)
```

Aufgrund eines Bugs in der whatis.library muß ein solcher "Füll"-Typ immer am Ende der Typen-Datei stehen, da sonst ein Typ zu wenig in der Liste geführt wird. Man sollte also folgende Zeilen ans Ende von "S:FileTypes" hängen:

```
TYPE "Foo"
  NAMEPATTERN  "~(#?)"
ENDTYPE
```

Das dazu. Als letztes noch die Besprechung der Menüpunkte des ChangeIcon-Fensters:

Typen neu einlesen      Wenn man die Datei "S:FileTypes" geändert hat, treten die Änderungen erst nach Wählen dieses Menüpunkts in Kraft. Wenn das Neueinlesen nicht möglich ist, weil z. B. ein anderes Programm die whatis.library noch benutzt, blinkt der Bildschirm auf.

Typen editieren      Ruft einen Texteditor zum Ändern von "S:FileTypes" auf. Der Name des Editors kann über den ToolType

- » EDITOR
- angegeben werden. Fehlt dieser, wird die Environment-Variable "EDITOR" benutzt. Fehlt auch diese, wird "C:Ed" genommen. Nach Beenden des Editors wird automatisch versucht, die Typen neu einzulesen.

AppIcon / AppMenuItem	Entspricht dem gleichnamigen Schalter.
Über...	Zeigt Programinfos an.
Verlassen	Beendet ChangeIcon.

## 1.8 Der Rest

Ein paar abschließende Sachen. Zunächst die Danksagungen:

- Gérard Cornu für die ganzen französischen Übersetzungen
- André Voget und Tim Jöllenbeck für's Betatesten
- John Harper für den genialen JEd-Editor
- Rune Gram-Madsen für ASM-One V1.02, mit dem das Programm entstand
- ~Fred Fish für seine tolle FD-Serie
- Commodore für den Amiga, den Installer und AmigaGuide

Und die wahnsinnig lange und interessante History:

V1.0 (02.-27. Februar 1994) - es läuft

Tja. Was man noch einbauen könnte, wären ein möglicher Shell-Start und die Möglichkeit, Dateien per Filerequester auswählen zu können (so wie bei MegaView; tolles Programm übrigens!)... im Moment bin ich aber zu faul, aber vielleicht schaffst Du es ja, mich zu überreden, oder hast andere Vorschläge. Siehe dazu

» Meine Adresse

.

Also, viel Spaß mit dem Programm!!

## 1.9 Adresse des Autors

Wenn Du Bugreporte, Vorschläge oder eine übersetzte ↔

» CatalogTranslation-Datei

hast oder irgendwas anderes von mir willst, würde ich mich über einen Brief von Dir freuen:

Hanns Holger Rutz  
 Auf dem Godenstedter Berg 17  
 27404 Zeven  
 Tel. (0 42 81) 42 45

## 1.10 Workbench Applikationen

Ab OS2.0 gibt es drei Arten von Workbench Applikationen: AppIcons, AppMenuItems und AppWindows.

- AppIcons sind Piktogramme, die ein Programm im Workbench-Fenster installiert. Man kann sie doppelklicken oder andere Piktogramme auf sie ziehen.
- AppMenuItems sind Menüpunkte, die im "Hilfsmittel"-Menü der Workbench erscheinen. Man kann sie einfach anwählen oder vorher beliebig viele Piktogramme auf der Workbench selektieren.
- AppWindows sind Fenster, die auf dem Workbench-Bildschirm erscheinen. Man kann wie bei AppIcons Piktogramme hineinwerfen, mit denen das jeweilige Programm dann arbeiten kann. Ein Beispiel dafür ist das IconEdit-Programm, welches sich auf den Disketten Deines Workbench-Sets befindet.

## 1.11 Changelcon übersetzen

Wenn Du mir helfen und ChangeIcon in eine weitere Sprache ↔ übersetzen willst, benötigst Du lediglich einen ASCII-Editor und die Datei "ChangeIcon.ct". Diese enthält alle Texte auf englisch. Hier ein Beispiel:

```
; ** Req-text, LineFeeds allowed
MSG_NO_MEM
?
; Not enough free memory!
```

Die erste Zeile ist ein Kommentar, der weitere Information zum jeweiligen Text gibt. In diesem Fall wird gesagt, daß der Text in einem Requester erscheint und LineFeeds ("\n") enthalten darf. Die zweite Zeile ist nur für mich wichtig. Das Fragezeichen in der dritten Zeile muß Du durch die Übersetzung des englischen Textes ersetzen. Dieser steht als Kommentar in der vierten Zeile.

Wenn Du eine Übersetzung gemacht hast, schick mir die geänderte .ct-Datei einfach auf Diskette zu, und Du erhältst die fertige Catalog-Datei von mir (

» Adresse des Autors

).

## 1.12 ToolTypes

ToolTypes sind Parameter, die über die Workbench editiert werden können. Dazu klickt man ein Piktogramm, in diesem Fall das von ChangeIcon, an und wählt "Informationen ..." aus dem "Piktogramm" Menü der Workbench. In der "Merkmale" Liste sind alle ToolTypes aufgeführt. Diese kann man mit den Schaltern "Neu" und "Löschen" erweitern oder löschen. Zum Editieren klickt man den gewünschten ToolType einfach in der Liste an.

Ein ToolType besteht aus dem sogenannten Keyword, einem Schlüsselwort also. Bei manchen folgt ein Gleichheitszeichen und eine Option oder ein Text. Beispiel:

```
EDITOR=SYS:Tools/MEmacs
```

EDITOR ist das Keyword, Groß- und Kleinschreibung ist hier nicht relevant. Nach dem Gleichheitszeichen steht in diesem Beispiel der Pfad zu einem Texteditor. Gibt es mehrere festgelegte Optionen, werden sie in dieser Anleitung durch Pipes ("|") getrennt:

```
POPUP=YES|NO
```

Man kann also entweder "YES" oder "NO" hinter dem Keyword angeben.

Die Änderungen an den ToolTypes werden durch Anwählen von "Speichern" wirksam.

## 1.13 Eingebaute Typen

Folgende Typen beinhaltet die whatis.library bisher. Die Piktogramm-Namen lauten jeweils "def\_<TypName>", also zum Beispiel "def\_Text" für den "TEXT"-Typen.

8SVX	ein IFF/8SVX Sound
ANIM	eine IFF/ANIM Animation
Assign	Assign
Dir	Verzeichnis / Ordner; Achtung: der Piktogrammname lautet "def_Drawer"!
Exe	ausführbares Programm; Achtung: der Piktogrammname lautet "def_Tool"!
FTXT	eine IFF/FTXT Text-Datei
Icon	eine Piktogramm-Datei; steht nicht zur Verfügung!
IFF	eine IFF-Datei
ILBM	ein IFF/ILBM Bild
ILBM24	ein IFF/ILBM Bild mit 16,8 Mio Farben
Imp Data	eine mit dem Programm "Imploder" gepackte Daten-Datei
LHArc	ein mit dem Programm "LHArc" oder "LHa" gepacktes Archiv
Lib	eine Library (= Funktionsbibliothek; meist in "SYS:Libs")
Med MOD	ein mit dem Musikprogramm "MED" erstelltes Musikstück
Object	Object-Modul (finden beim Programmieren Verwendung)
PP Data	eine mit dem PowerPacker gepackte Daten-Datei
PP Exe	ausführbares Programm, mit PowerPacker gepackt
PP30 Exe	ausführbares Programm, mit PowerPacker 3.0 gepackt
PP40 Exe	ausführbares Programm, mit PowerPacker 4.0 gepackt
PREFS	eine IFF/PREFS Datei, wird von den Voreinstellern benutzt
Pure Exe	ausführbares Programm, das resident gemacht werden kann (erkennbar am gesetzten "P"-Bit)
Script	Batch-Datei für die Shell (erkennbar am gesetzten "S"-Bit)
SMUS	ein IFF/SMUS Musikstück
TERM	eine Datei vom Programm "Term"
Text	reine Text-Datei
Volume	Datenträger; Achtung: der Piktogrammname lautet "def_Disk"!
Zoo	ein mit dem Programm "Zoo" gepacktes Archiv

## 1.14 Index

Zu faul... :-)

Was anderes: Kennst Du das deutsche, frei kopierbare Disketten-Magazin mit Namen "Forum Amiga"? Nein, solltest Du aber. Denn es handelt sich dabei um ein Magazin von Freaks für Freaks - ohne kommerzielle Werbung. Es bietet Themen für jeden - Programmierkurse, FD-Tests, Tests kommerzieller Software und Hardware, Berichte, Leserforum, Tips+Tricks, Film- und CD-Vorstellungen und anderes. Interessiert? Dann schick' mir einfach eine Leerdiskette und einen ausreichend frankierten Rückumschlag, und ich kopier' Dir die neuste Ausgabe des Forums. Es lohnt sich!

---