

## **Backprefs**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> Backprefs		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 26, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Backprefs</b>	<b>1</b>
1.1	Background Voreinstellungseeditor . . . . .	1
1.2	Copyright, Vertrieb ... . . . .	1
1.3	Benötigte libraries . . . . .	2
1.4	fehler . . . . .	2
1.5	Programmversionen bis heute . . . . .	2
1.6	Geplante Verbesserungen . . . . .	2
1.7	Ich selbst . . . . .	2
1.8	Welche Bilder kann ich verwenden? . . . . .	3
1.9	Background preferences . . . . .	3
1.10	Das Symbol 'Name' . . . . .	4
1.11	Das Symbol 'Pattern' . . . . .	4
1.12	Das Symbol 'Task' . . . . .	4
1.13	Das TILE-Flag . . . . .	5
1.14	Das Screen-Flag . . . . .	5
1.15	Delay loading . . . . .	5
1.16	Das Diskettensymbol . . . . .	5
1.17	Die Bilderliste . . . . .	6
1.18	Das Symbol 'Add' . . . . .	6
1.19	Das Symbol 'Del' . . . . .	6
1.20	Das Symbol 'Up' . . . . .	6
1.21	Das Symbol 'Down' . . . . .	6
1.22	Das Symbol 'View' . . . . .	6
1.23	Das Symbol 'Save' . . . . .	7
1.24	Das Symbol 'Use' . . . . .	7
1.25	Das Symbol 'Cancel' . . . . .	7
1.26	Das Informationfeld . . . . .	7
1.27	Die Menüs . . . . .	7
1.28	Project/Open . . . . .	8
1.29	Project/Save As . . . . .	8

---

---

1.30 Project/About . . . . .	8
1.31 Project/Quit . . . . .	8
1.32 Edit/Reset To Defaults . . . . .	8
1.33 Edit/LastSaved . . . . .	8
1.34 restore . . . . .	8
1.35 Picture/Compute Colors . . . . .	9
1.36 algo . . . . .	9
1.37 Picture/Dither . . . . .	11
1.38 Picture/Lock Colors . . . . .	11
1.39 Picture/Change Colors . . . . .	11
1.40 Picture/Load Colors . . . . .	11
1.41 Picture/Save Selected . . . . .	11
1.42 Settings/SaveIcons? . . . . .	11
1.43 median . . . . .	12
1.44 custom . . . . .	12

---

# Chapter 1

## Backprefs

### 1.1 Background Voreinstellungseditor

Backprefs v2.0

Backprefs und dieser Text sind Copyright ©1996 Roland Haas

Inhalt:

- Rechtliches
- Benötigte libraries
- Bekannte Fehler
- Geschichte
- Zukünftiges
- Autor

- Die Bilder
- Die Gadgets
- Die Menüs

### 1.2 Copyright, Vertrieb ...

Vertrieb

Backprefs darf frei vertrieben werden, solange der Inhalt des Archivs komplett weitergegeben wird. Außerdem darf nur eine Gebühr für den Arbeits- und Materialaufwand verlangt werden.

Backprefs darf mit frei kopierbarer Software vertrieben werden, solange die oben genannten Bedingungen erfüllt sind. Der Vertrieb zusammen mit einem kommerziellen Programm ist nur mit ausdrücklicher, schriftlicher Genehmigung des Autors erlaubt.

Gewährleistung

Es wird keine Haftung für Schäden übernommen, die dieses Programm verursacht. Es wird keine Gewähr dafür übernommen, daß das Programm im

Sinne dieses Textes anwendbar ist.

SIE BENUTZEN DIESES PROGRAMM AUF EIGENES RISIKO

INSTALLER SOFTWARE IS PROVIDED "AS-IS" AND SUBJECT TO CHANGE;  
NO WARRANTIES ARE MADE. ALL USE IS AT YOUR OWN RISK. NO LIABILITY  
OR RESPONSIBILITY IS ASSUMED.

Installer and Installer project icon

(c) Copyright 1995-96 Escom AG. All Rights Reserved.

Reproduced and distributed under license from escom AG.

## 1.3 Benötigte libraries

Das Voreinstellungsprogramm benötigt die 'gadutil.library', die sich im Verzeichnis LIBS: befinden muß. Das Install-Script sollte das erledigen.

## 1.4 fehler

Momentan sind mir keine Fehler bekannt.

## 1.5 Programmversionen bis heute

Achtung: Die Versionsnummer beginnt bei Version 2.0. Das Programm erhält immer dieselbe Versionsnummer wie das Hauptprogramm. Nur Programme gleicher Version funktionieren miteinander!

03.08.1996: V2.0 erste Version  
08.04.1997: V2.1 Farbanpassung  
18.05.1997: V2.11 Einige Fehler entfernt

## 1.6 Geplante Verbesserungen

- Aufruf mit Parametern

Vorschläge, Kritik, Fehler bitte an den Autor

## 1.7 Ich selbst

Backprefs wurde in harter Nachtarbeit geschrieben von:

Roland Haas  
Mühlweg 16  
77948 Friesenheim

---

Telefon: 07821/68084

Die Benutzeroberfläche wurde erstellt GadUtilLib v37.

Anregungen, Spenden, Postkarten sind immer willkommen!  
 Besonders willkommen wäre eine bessere Version dieses Textes, sowie  
 Übersetzungen der Anleitung...

## 1.8 Welche Bilder kann ich verwenden?

Es können nur Bilder im IFF-ILBM-Format geladen werden. Die Bilder sollten dasselbe Seitenverhältnis haben wie Ihre Workbench. Wenn Ihre Workbench im Modus HIRES/INTERLACED läuft, sollte das Bild auch diesen Bildschirmmodus verwenden oder LORES/NONLACED. Bedenken Sie bei der Auswahl Ihrer Bilder immer, daß die Bilder ungepackt im Speicher vorliegen, ein 640x320 Punkte großes Bild mit acht Farben braucht also  $640 \times 320 \times 3/8 = 76800$  Byte Chip-Ram. Außerdem müssen die Bilder zum Programmstart geladen werden; bei vielen großen Bildern kann das schon mal ein paar Sekunden dauern. Verwenden Sie daher lieber wenige kleine Bilder und setzen das TILE-Flag als viele große Bilder. Ab Version 2.1 können zwar auch Bilder während der Laufzeit nachgeladen werden, das benötigt jedoch mehr oder weniger viel Zeit.

## 1.9 Background preferences

					Loaded Pictures:
	+	-----+	+	+	+
Name		Work:Pictures/Diskette		?	Back1  #
	+	-----+	+	+	+
	+	-----+			Romantique  #
	+	-----+			Diskette  #
Pattern		"(#? voll, #? frei			Wüste  #
	+	-----+			^
	+	-----+			V
Task		WORKBENCH			+
	+	-----+			Diskette
	+	-----+			+
		View			Add  Del   Up  Down
	+	-----+			+
	+	----- Information -----+			+
		Picture Workbench			Tile picture:  V
					+
		Size: 96 x 96 640 x 256			+
					Screen picture:
		Planes: 3 3			+
					+
		Aspect: 10 : 22 22 : 44			Delay loading:
	+	-----+			+
	+	-----+			+
		Save			Cancel
	+	-----+			+
	+	-----+			+

## 1.10 Das Symbol 'Name'

Hier wird der Name des Bildes eingegeben. Das Symbol daneben öffnet einen Dateiauswahlfenster für den gleichen Zweck. Der hintere Teil des Namens erscheint in der Bilderliste.

## 1.11 Das Symbol 'Pattern'

Hier wird das Fenstermuster angegeben, das für dieses Bild gilt. Fenster, deren Titel auf dieses Muster paßt verwenden dieses Bild als Hintergrundmuster.

Beispiele:

```
"#? voll, #? frei, #? belegt" - paßt für alle Disketten
(3D|DPAINT|XiPaint)          - für die Schubladen 3D, DPAINT und
                               XiPaint
```

Mögliche Musterzeichen:

```
?      Ein einzelnes Zeichen
#      Das nächste Symbol beliebig oft (auch 0 mal)
(ab|cd) Eins der durch '|' getrennten Symbole
~      Negiert den folgenden Ausdruck. D. h. ~(System) paßt
       auf alles außer 'System'
[abc]  Buchstabenmenge, paßt auf jeden Buchstaben der Menge
[~bc]  Buchstabenmenge, paßt auf alle Buchstaben, die nicht in
       der Menge enthalten sind
a-z    Buchstabenraum (nur in Buchstabenmengen)
%      Nichts, nützlich in 'foo(bar|%)'
```

## 1.12 Das Symbol 'Task'

Hier wird das Taskmuster angegeben, das für dieses Bild gilt. Programme, deren Titel auf dieses Muster paßt verwenden dieses Bild als Hintergrundmuster für ihre Fenster.

Beispiele:

```
WORKBENCH      - paßt nur für die Workbench
#?DeliTracker#? - paßt für den Delitracker
```

Die beiden '#?' Zeichen um das Muster sind nützlich, denn aus 'DeliTracker' kann 'dh0:Tools/DeliTracker' werden wenn er aus dem CLI heraus aufgerufen wird.

Mögliche Musterzeichen:

```
?      Ein einzelnes Zeichen
#      Das nächste Symbol beliebig oft (auch 0 mal)
(ab|cd) Eins der durch '|' getrennten Symbole
~      Negiert den folgenden Ausdruck. D. h. ~(System) paßt
       auf alles außer 'System'
[abc]  Buchstabenmenge, paßt auf jeden Buchstaben der Menge
```



```
[~bc] Buchstabenmenge, paßt auf alle Buchstaben, die nicht in
      der Menge enthalten sind
a-z   Buchstabenraum (nur in Buchstabenmengen)
%     Nichts, nützlich in 'foo(bar|%)'
```

### 1.13 Das TILE-Flag

Das Symbol 'Tile Picture' setzt oder löscht für das aktive Bild das TILE-Flag. Ist dieses Flag gesetzt, so wird das Bild solange aneinandergesetzt, bis das Fenster gefüllt ist. Ansonsten wird der restliche Bereich mit Farbe 0 (grau) gelöscht. Damit das TILE-Flag funktioniert müssen die Kanten des Bildes aneinanderpassen.

### 1.14 Das Screen-Flag

Das Symbol 'Screen Picture' setzt oder löscht für das aktive Bild das Screen-Flag. Ist dieses Flag gesetzt, dann kann dieses Bild dazu verwendet werden, um im Hintergrund eines Bildschirms angezeigt zu werden. Das Muster wird verwendet, um festzustellen, für welchen Screen das Bild passt. Dabei gilt der Titel, der angezeigt wird, wenn kein Fenster aktiv ist. Um den Bildschirm der Workbench zu benutzen, müßten die Einträge folgendermaßen aussehen:

```
Pattern: "Workbench Screen"
Task: "WORKBENCH"
ScreenPic: gesetzt
```

### 1.15 Delay loading

Das Symbol 'Delay loading' bestimmt, wann das Bild geladen werden soll. Wenn es gesetzt ist, dann wird das Bild erst dann geladen, wenn es zum ersten Mal benötigt wird. Wenn später einmal Speicher gebraucht wird, dann wird das Bild, sofern es nicht gebraucht wird, wieder freigegeben. Wer viele Bilder verwenden will, der sollte bei denjenigen, die nicht so oft gebraucht werden, 'Delay loading' verwenden, sonst braucht Background zu viel Zeit, um sie alle zu laden.

### 1.16 Das Diskettensymbol

Dieses Symbol öffnet ein Dateiauswahlfenster, mit dem ein Bild ausgewählt werden kann. Der Pfad erscheint dann im Texteingabefeld daneben und in der Bilderliste.

## 1.17 Die Bilderliste

Im rechten Teil des Fensters werden alle momentan geladenen Bilder in einer Liste angezeigt. Das jeweils aktive Bild wird unter der Liste angezeigt. Pfad, Fenstermuster, {"Taskmuster" LINK Task} und sonstige Attribute des aktiven Bildes können mit Hilfe der anderen Bedienelemente verändert werden.

## 1.18 Das Symbol 'Add'

Das Add-Symbol erzeugt einen neuen Eintrag für ein Bild. Pfad und Muster werden aus den entsprechenden Texteingabefeldern übernommen. Der Typ des neuen Bildes wird auf 'Normal' gesetzt, das TILE-Flag gesetzt. Das neue Bild wird entweder nach dem aktiven Bild oder am Ende der Liste eingefügt.

## 1.19 Das Symbol 'Del'

Das Symbol Delete löscht das aktive Bild aus der Liste; die Texteingabefelder bleiben jedoch unverändert. Wenn Sie also das falsche Bild gelöscht haben, können Sie es mit Hilfe des Add-Symbols daneben wiederherstellen.

## 1.20 Das Symbol 'Up'

Das Symbol Up bewegt das aktive Bild einen Platz weiter nach oben. Wenn auf ein Fenster bzw. einen Bildschirm zwei Bilder passen, dann wird dasjenige verwendet, das sich weiter unten befindet. Das Bild mit dem Muster '#?' sollte also ganz oben stehen, weil sonst die anderen nicht verwendet werden.

## 1.21 Das Symbol 'Down'

Das Symbol Down bewegt das aktive Bild einen Platz weiter nach unten. Wenn auf ein Fenster bzw. einen Bildschirm zwei Bilder passen, dann wird dasjenige verwendet, das sich weiter unten befindet. Unten sollten sich also nur Bilder befinden, die nicht auf mehrere Fenster passen, vor allem aber keine Bilder mit einem Fenstermuster wie '#?'.

## 1.22 Das Symbol 'View'

Durch Anklicken des Symbols View öffnet sich ein Fenster, in dem das aktive Bild so dargestellt wird, wie es später im Workbenchfenster zu sehen sein wird. Um Änderungen (z. B. des TILE-Flags) sichtbar zu machen müssen Sie das Symbol erneut anwählen.

---

## 1.23 Das Symbol 'Save'

Save beendet das Programm, lädt die neuen Bilder und speichert die Liste dauerhaft, sodaß sie nach einem Neustart wieder verwendet wird.

## 1.24 Das Symbol 'Use'

Use beendet das Programm, lädt die neuen Bilder und speichert die Liste sodaß sie bis zum nächsten Neustart zur Verfügung steht.

## 1.25 Das Symbol 'Cancel'

Cancel beendet das Programm ohne die Bilderliste zu speichern.

## 1.26 Das Informationfeld

Im Informationsfeld werden Daten über die Geometrie des Bildes und der Workbench ausgegeben. Size gibt die Größe des Bildes bzw. des Bildschirms an. Wenn das Bild größer als der Workbenchbildschirm ist, wird ein Teil des Bildes niemals sichtbar sein. Planes gibt die Anzahl der Bitplanes an, die verwendet werden. Die Anzahl der Farben berechnet sich folgendermaßen:

$$2^{\text{Anzahl der Bitplanes}}$$

Wenn Sie also 3 Bitplanes verwenden stehen Ihnen 8 Farben zur Verfügung. Hat das Bild mehr Farben als Ihre Workbench, dann wird das Bild wahrscheinlich nicht so aussehen wie es sollte. Verwenden Sie also nur Bilder, die gleich viel oder weniger Farben haben als Ihre Workbench. Der letzte Wert, Aspect, gibt das Verhältnis von Höhe und Breite wieder. Es sollte bei Workbench und Bild etwa übereinstimmen, da es sonst zu Verzerrungen kommt. Also z. B. 11:22 und 10:20, oder 11:11 und 10:12 etc.

Welche Bilder kann Ich verwenden?

## 1.27 Die Menüs

Die Menüs entsprechen exakt denen der Systemvoreinsteller, sodaß Sie zu Ihrer Funktion auch im Systemhandbuch nachschlagen können, falls Sie etwas nicht verstehen.

Project	Edit	Picture	Settings
Open ...	Reset To Defaults	Compute Colors	Save Icons?
Save As ...	Last Saved	Dither	Quantization >>
About ...	Restore	Lock Colors ...	Median Cut
Quit		Change Colors ...	Custom
		Load Colors ...	
		Save Selected ...	

## 1.28 Project/Open

Lädt eine Datei mit einer Bilderliste.

## 1.29 Project/Save As

Läßt Sie die momentane Bilderliste in eine Datei speichern. Es öffnet sich ein Dateiauswahlfenster, das standartmäßig das Verzeichnis Sys:Prefs/Presets/ als Ziel vorsieht. Ist 'Save Icons?' ausgewählt, dann wird auch ein Piktogramm erzeugt. Im Gegensatz zu den Systemvoreinstellern wird durch Öffnen dieses Piktogramms nicht die Bilderliste geladen, sondern nur das Voreinstellungsprogramm gestartet. Verwenden Sie den Menüpunkt Project/Open um die Bilderliste zu laden.

## 1.30 Project/About

Dieser Menüpunkt zeigt Daten über das Programm an. (U. a. Version, Adresse des Autors, Hilfsmittel...)

## 1.31 Project/Quit

Verlassen des Editors ohne Speichern der Änderungen. Dieses Option hat die gleiche Funktion wie 'Cancel'.

## 1.32 Edit/Reset To Defaults

Setzt die Einstellungen des Editors auf die Standarteinstellungen zurück. Genauer bedeutet das, daß wieder die Muster verwendet werden, die mit WBPATTERN festgelegt wurden.

## 1.33 Edit/LastSaved

Lädt die Einstellungen, die zuletzt mit 'Save' gespeichert wurden.

## 1.34 restore

Zurücksetzen der Einstellungen auf die Einstellungen beim ersten öffnen des Editors.

---

## 1.35 Picture/Compute Colors

Versucht eine Farbpalette zu finden, die für alle Bilder der Bilderliste passt. Es stehen zwei verschiedene Methoden der Farbauswahl zur Verfügung. Median Cut ist ein Verfahren nach Paul S. Heckbert. Custom ist eine Eigenentwicklung meinerseits. Im Gegensatz zum Median Cut können dabei jedoch einzelne Farben gesperrt werden, sodaß sie nicht verändert werden. Allerdings weiß ich nicht, ob mein Verfahren sehr gut oder schnell ist. Wer ein bißchen Ahnung davon hat, soll sich doch einmal den Algorythmus durchlesen und mir seine Meinung dazu schicken.

## 1.36 algo

1. Schritt: Alle Farben, die in den Bildern vorkommen, werden gezählt
2. Schritt: Die Farben werden in die zur Verfügung stehenden Farbtöpfe eingeordnet. Ist kein freier Topf mehr vorhanden, werden die zwei einander ähnlichsten zusammengefasst. Dabei wird auch die neue Farbe mitberücksichtigt.
3. Schritt: Schließlich wird die Farbe für das Farbregister bestimmt, indem für jeden Topf die Farbwerte zusammengezählt werden und der Durchschnitt gebildet wird.

Der erste und der dritte Schritt sind auf jeden Fall in Ordnung, hier der Quellcode für den zweiten:

```
/* Colourcount enthält die Anzahl der gefundenen Farben */

for( i = ColourCount ; i > 0 ; i--)
{

    /* Einen freien Farbtopf suchen */

    NewPot = NULL;
    for(j = NumColours ; j > 0 ; j--)
    {
                                                /* Der Farbtopf ist noch leer
                                                * (gesperrte Farben haben hier 1)
                                                */
        if (Pot[j] . NumPixels == 0)
        {
            NewPot = &Pot[j];
            break;
        }
    }

    if(NewPot != NULL) /* Wir haben noch einen freien gefunden */
    {
        /* sRed, sGreen, sBlue enthalten die zusammenaddierten
           Farbwerte; Red, Green, Blue die endgültigen
           */

        pix = NewPot -> NumPixels = ColourTable[i-1] . NumPixels;
        NewPot -> sRed   = (NewPot -> Red   = ColourTable[i-1] . Red   ) * pix;
        NewPot -> sGreen = (NewPot -> Green = ColourTable[i-1] . Green) * pix;
```

```
NewPot -> sBlue = (NewPot -> Blue = ColourTable[i-1] . Blue )*pix;
}
else
{
    /* So, jetzt machen wir einen Topf leer */

    pix = Pot[0] . NumPixels = ColourTable[i] . NumPixels;
    Pot[0] . sRed = (Pot[0] . Red = ColourTable[i-1] . Red )*pix;
    Pot[0] . sGreen = (Pot[0] . Green = ColourTable[i-1] . Green)*pix;
    Pot[0] . sBlue = (Pot[0] . Blue = ColourTable[i-1] . Blue )*pix;

    Dist = 0x7fffffff; /* Größtmöglicher Abstand */
    for(j = NumColours ; j > 0 && Dist; j--)
    {
        for(k = j - 1 ; k >= 0 && Dist ; k--)
        {
            DistRGB = Pot[k] . Red - Pot[j] . Red;
            NewDist = DistRGB * DistRGB;

            DistRGB = Pot[k] . Green - Pot[j] . Green;
            NewDist += DistRGB * DistRGB;

            DistRGB = Pot[k] . Blue - Pot[j] . Blue;
            NewDist += DistRGB * DistRGB;

            /* Wenn die Farbe näher dran ist als die alte, und nicht
               beide Farben gesperrt sind
            */

            if ((NewDist <= Dist) && !(Pot[k] . Locked && Pot[j] . Locked))
            {
                NewPot = &Pot[j];
                NewPot2 = &Pot[k];
                Dist = NewDist;
            }
        }
    }

    /* Die zwei zusammenfassen */

    if (!(NewPot -> Locked) || (NewPot2 -> Locked))
    {
        pix = (NewPot -> NumPixels += NewPot2 -> NumPixels);
        NewPot -> Red = ((NewPot -> sRed += NewPot2 -> sRed)/pix);
        NewPot -> Green = ((NewPot -> sGreen += NewPot2 -> sGreen)/pix);
        NewPot -> Blue = ((NewPot -> sBlue += NewPot2 -> sBlue)/pix);
    }
    else if (NewPot2 -> Locked)
    {
        NewPot -> Red = NewPot2 -> Red;
        NewPot -> Green = NewPot2 -> Green;
        NewPot -> Blue = NewPot2 -> Blue;
    }

    /* Die neue Farbe richtig einfügen */
}
```

```
        *NewPot2 = Pot[0];  
    } /* else */  
  
} /* for i */
```

### 1.37 Picture/Dither

Passt das aktive Bild an die aktuelle Farbpalette an. Dabei wird der Floyd-Steinerg-Algorithmus verwendet.

### 1.38 Picture/Lock Colors

Es erscheint ein Fenster, in dem Farben gesperrt werden können. Gesperrte Farben werden nicht verändert wenn Compute Colors aufgerufen wird. Nur die Custom Methode unterstützt gesperrte Farben, Median Cut nimmt keine Rücksicht auf sie.

### 1.39 Picture/Change Colors

Es erscheint ein Fenster, in dem die Farben geändert oder umgestellt werden können. Das Fenster ist zu bedienen wie gebräuchliche Farbrequester.

### 1.40 Picture/Load Colors

Mit Hilfe dieses Menüpunktes können Sie die Farben einer IFF Datei laden. Wenn Sie also ihre Bilder an die Farben der Workbench anpassen wollen, dann laden Sie die Datei 'ENV:Sys/Palette.ilbm' und wählen dann Dither für jedes Bild, das auf der Workbench zu sehen sein wird. Um die Farben von andren Bildschirmen zu verwenden, müssen Sie einen Screenshot davon machen und dann diese Datei laden.

### 1.41 Picture/Save Selected

Mit Hilfe dieses Menüpunktes können Sie das gerade aktive Bild speichern. Es erscheint ein Dateiauswahlfenster, in dem Sie angeben müssen, wohin Sie das Bild speichern wollen.

### 1.42 Settings/Savelcons?

Mit Hilfe dieses Menüpunktes können Sie Projektpiktogramme für die Einstellungen des Editors in denselben Schubladen speichern, in denen sich auch die zugehörigen Daten befinden. Die Aktivierung der Einstellungen durch Öffnen des Piktogramms ist jedoch nicht möglich. Stattdessen müssen Sie Project/Open benutzen.

## 1.43 median

Ist dieser Menüpunkt angewählt, dann wird für die Farbanpassung der Median Cut Algorithmus verwendet. Diese Methode erlaubt es nicht, einzelne Farben zu sperren.

## 1.44 custom

Ist dieser Menüpunkt angewählt, dann wird für die Farbanpassung ein von mir selbst geschriebener Algorithmus verwendet. Diese Methode erlaubt es, einzelne Farben zu sperren.

---