Backprefs

COLLABORATORS			
	TITLE : Backprefs		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 5, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Backprefs **1**

1.1 Background preferences editor . . . 1

1.2 Copyright, copying 1

1.3 fehler . . . 2

1.4 Versions until today . . . 2

1.5 Planned features . . . 2

1.6 This is me . . . 2

1.7 Which pictures can be used? . . . 3

1.8 Background preferences . . . 3

1.9 The 'Name' gadget . . . 4

1.10 The 'Pattern' gadget . . . 4

1.11 The 'Task' gadget . . . 5

1.12 The TILE flag . . . 5

1.13 The Screen flag . . . 6

1.14 Delay loading . . . 6

1.15 The floppy symbol . . . 6

1.16 The list of pictures . . . 6

1.17 The 'Add' gadget . . . 7

1.18 The 'Delete' gadget . . . 7

1.19 The 'Up' gadget . . . 7

1.20 The 'Down' gadget . . . 8

1.21 The 'View' gadget . . . 8

1.22 The 'Save' gadget . . . 8

1.23 The 'Use' gadget . . . 8

1.24 The 'Cancel' gadget . . . 9

1.25 Given informations . . . 9

1.26 The menus . . . 9

1.27 Project/Open . . . 10

1.28 Project/Save As . . . 10

1.29 Project/About . . . 10

1.30 Project/Quit 10
1.31 Edit/Reset To Defaults 11
1.32 Edit/LastSaved 11
1.33 restore 11
1.34 Picture/Compute Colors 11
1.35 algo 12
1.36 Picture/Dither 13
1.37 Picture/Lock Colors 14
1.38 Picture/Change Colors 14
1.39 Picture/Load Colors 14
1.40 Picture/Save Selected 14
1.41 Settings/SaveIcons? 14
1.42 median 15
1.43 custom 15

Chapter 1

Backprefs

1.1 Background preferences editor

Backprefs v2.1

Backprefs and this text are Copyright ©1996 Roland Haas

Contents:

Disclaimer

Known bugs

History

Future

Author

Which pictures

The gadgets

The menus

1.2 Copyright, copying ...

Copying

Backprefs may be distributed freely, as long as the contents of the archive remains unchanged. No charge may be raised other then a small fee for copying.

Backprefs may be sold with other free redistributable software, as long as the conditions above are satisfied. This program may only be included in a commercial package if you have the written permission of the ‘author.

disclaimer

This product is sold "AS IS" without representation or warranty of any kind, either expressed or implied.

YOU USE THIS PROGRAM ON YOUR OWN RISK

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.

1.3 fehler

Currently no bugs are known to me.

1.4 Versions until today

NOTE: The version numbering starts at version 2.0. This program always gets the same version number as th main program background. Using programs with different version numbers will probably crash the machine.

03.08.1996: V2.0 initial version
08.04.1997: V2.1 color adaption

1.5 Planned features

- Supporting parameters
- slider in preview window

Suggestions, bug reports to the
Author

1.6 This is me

Backprefs was coded during long nights by:

Roland Haas
Mühlweg 16
77948 Friesenheim

phone: ++49 07821/68084

the GUI was created with GadToolsBox v2.0.

Suggestion, gifts and post cards are always welcome!
A new and better version of this text would be especially welcome. Perhaps
even a translation???

1.7 Which pictures can be used?

Only pictures in the IFF-ILBM format can be loaded. The pictures _must_ use the same palette as the workbench. The pictures should have the same aspect ratio. E.g. if your workbench is HIRES/INTERLACED the picture should use the same viewmode or LORES/NONLACE. Always keep in mind that the pictures remain in memory; a 640x320 sized picture using eight colors will occupy 640*320*3/8 = 76800 byte CHIP-RAM. Also pictures must be loaded at program startup; if you are using many big pictures, this can take several seconds. So better use few little pictures setting the TILE flag than many bigger ones. Starting with V2.1 you can load pictures later, but this takes of course some time.

1.8 Background preferences

```
                Loaded Pictures:
                          +---------------------------------+ +-+  ↩
                             +-----------------+-+

                   Name
                 | Work:Pictures/Disc             | |
                ?
                | | Back1          |#|
          +---------------------------------+ +-+ | Romantique      |#|
          +---------------------------------+     | Disc            |#|

                Pattern
                 | "(#? full, #? free             |     | Desert           |#|
          +---------------------------------+     |                 |^|
          +---------------------------------+     |                 |V|

                   Task
                 | WORKBENCH                      |     +-----------------+-+
          +---------------------------------+     | Disc              |
          +------------+                          +----+----+----+----+
          |
                    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
                |                |
                     Use
                |                   |
                   Cancel
                |
  +------------+                +------------+                  +------------+
```

1.9 The 'Name' gadget

You enter the path to the picture here. The button beside opens a filerequester for doing the same job. The last part of the path appears in the list of pictures .

1.10 The 'Pattern' gadget

Here you can enter the pattern, which is to use for this window. Windows, whose titles match the pattern will use this picture as their background

pattern.
Examples:

```
   "#? full, #? free, #? in use" - matches all discs
   (3D|DPAINT|XiPaint)           - matches the drawers 3D, DPaint and
                                   XiPaint
```

Possible tokens:

```
        ?       Matches a single character.
        #       Matches the following expression 0 or more times.
        (ab|cd) Matches any one of the items separated by '|'.
        ~       Negates the following expression.  It matches all strings
                that do not match the expression (aka ~(foo) matches all
                strings that are not exactly "foo").
        [abc]   Character class: matches any of the characters in the class.
        [~bc]   Character class: matches any of the characters not in the
                class.
        a-z     Character range (only within character classes).
        %       Matches 0 characters always (useful in "(foo|bar|%)").
```

1.11 The 'Task' gadget

Here you can enter the task pattern, which is to use for this window. Tasks, whose names match the pattern will use this picture as a background pattern for their windows.
Examples:

```
   WORKBENCH            - only matches the Workbench task
   #?DeliTracker#?       - the famous DeliTracker
```

Note the two '#?'s surounding the pattern. They are quite usefull, since 'DeliTracker' may become 'dh0:Tools/DeliTracker' when started from CLI.

Possible tokens:

```
        ?       Matches a single character.
        #       Matches the following expression 0 or more times.
        (ab|cd) Matches any one of the items separated by '|'.
        ~       Negates the following expression.  It matches all strings
                that do not match the expression (aka ~(foo) matches all
                strings that are not exactly "foo").
        [abc]   Character class: matches any of the characters in the class.
        [~bc]   Character class: matches any of the characters not in the
                class.
        a-z     Character range (only within character classes).
        %       Matches 0 characters always (useful in "(foo|bar|%)").
```

1.12 The TILE flag

The gadget 'Tile Picture' sets or clears the Tile flag of the active

picture. By setting this flag the picture will be tiled to cover the whole
area. Otherwise the remaining areas would be cleared with color 0 (grey).
For this flag to work well, the
edges of the picture must fit together
.

1.13 The Screen flag

The gadget 'Screen Picture' sets or clears the screen flag of the
active
picture. If this flag is set, the picture can be used to fill the blank
area on screens. The
pattern
is used to tell which screen should use the
picture. To match the Workbench screen you must use the following:

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

1.14 Delay loading

The gadget 'Delay loading' determines when the picture is loaded. If it is
set, then the picture is loaded when it is used for the first time. If
there is some memory needed later, this picture may be freed again to get
unused memory free. If you plan to use many pictures, you should set this
flags for those, that are rarely used. Background will need lots of time to
load them at startup otherwise.

1.15 The floppy symbol

This gadget will open a filerequester allowing you to select a picture. The
path appears then in the
string gadget
beside and in the
list of pictures
.

1.16 The list of pictures

In the right part of the window all currently loaded pictures are shown in a listview. The active picture is shown below the list. Path , window pattern , task pattern and other properties can be change using the remaining gadgets.

1.17 The 'Add' gadget

The Add gadget will create a new entry for a picture. Path and pattern are taken from the corresponding stringgadgets. The type of the new picture is set to 'Normal', the TILE flag is also set. The new picture is either added just after the active picture or at the end of the l List of pictures .

1.18 The 'Delete' gadget

The Delete gadget removes the active picture from the list of pictures ; however string gadgets remain untouched. So if you have by mistake deleted the wrong picture, regain it using the Add gadget beside.

1.19 The 'Up' gadget

The Up gadget moves the active picture

one position up.
If two pictures match the same window the one later in the
list
will be
used. So the picture with the '#?' pattern should be on top of the
list
.

1.20 The 'Down' gadget

The Down gadget moves the
active picture
one position down.
If two pictures match the same window the one later in the
list
will be
used. So there should only pictures at the end of the list that do only
match one window and especially no one with a '#?'
pattern
.

1.21 The 'View' gadget

Selecting View will open a window in which you can see the
active picture
like you would see it in an workbench window. To make changes (e.g ↩
. the

TILE flag
) take effect you will have to select the gadget again.

1.22 The 'Save' gadget

Save quits the editor, loads the pictures and saves the list so it is be
used after a reboot.

1.23 The 'Use' gadget

Use quits the editor, loads the new pictures and saves the list until the
next reboot.

1.24 The 'Cancel' gadget

Cancel quits the editor without saving the list.

1.25 Given informations

In this field information about size, aspect ratio and depth of the picture and your workbench are given. Size shows you (surprise, surprise) the Size of the picture or the workbench in pixels. Pictures that are bigger than your workbench will never be shown entirely. The number of colors is computed as follows:

2^number of bitplanes

So if you are using 3 bitplanes you can choose from eight colors. A picture using more colors than the workbench will probably look weird. Because of this you should always choose pictures having as many as or less colors than your workbench. The last value Aspect shows the ratio of height to width. Workbench and picture should have the same ratio; otherwise distortion will occur.

Which pictures can be used?

1.26 The menus

The menus are the same as the ones of the system preferences editors, so you can also look their functions up in your system manual, if you don't understand something.

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 ...

1.27 Project/Open

Loads a list of pictures from a file.

1.28 Project/Save As

Lets you save the current list of pictures into a file. A file ↩ requester
opens up showing Sys:Prefs/Presets as the default destination. If
Save

Icons?
is selected then an icon will be created to. Other than the ↩ system
editors you can’t load the list by double clicking on the icon. Instead you
will have to use
Project/Open
to load the list.

1.29 Project/About

This menu item shows information about the program. (
Version, address of

authors, utilities...
)

1.30 Project/Quit

Quits the editor without saving the list of pictures. This option ↩ has the same effect as Cancel.

1.31 Edit/Reset To Defaults

Clears the list of pictures. That means the patterns defined in WBPATTERN will be used.

1.32 Edit/LastSaved

Loads the last settings that were saved using Save.

1.33 restore

Resets the settings to the ones preset at program startup.

1.34 Picture/Compute Colors

Trys to find colors that match for all pictures in the list. There are two different methods for doing this.

Median Cut is a procedure apadted from Paul S. Heckbert.

Custom is something created by me. In contrast to Median Cut colors can be locked, so that they won't be changed. But I really don't know if it is very fast or good. If somebody knows something about such things, he may look at the algorithm and send his opinion to me.

1.35 algo

```
   1. Step: All colors that are used in the pictures are counted
   2. Step: the colors are sorted into the available color pots. If there
            is no free one, the two, which are the most similar to each
            other, are combined taking the new color into account.
   3. Step: Finally the color for every pot is calculated by summing up all
            color values calculating the average of them.
```

the first and the last step are ok, so here's the source for the second step:

```
         /* ColourCount contains the number of used colours */

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

            /* Search for some free pot */

         NewPot = NULL;
         for(j = NumColours ; j > 0 ; j--)
         {
                                          /* The pot is still empty
                                           * (locked colors have a 1)
                                           */
            if (Pot[j] . NumPixels == 0)
            {
               NewPot = &Pot[j];
               break;
            }
         }

         if(NewPot != NULL) /* we have found a free one */
         {
            /* sRed, sGreen, sBlue contained the summed up color values
               Red, Green, Blue the real ones
            */

            NewPot -> NumPixels = ColourTable[i-1] . NumPixels;
            NewPot -> Red   = NewPot -> sRed   = ColourTable[i-1] . Red;
            NewPot -> Green = NewPot -> sGreen = ColourTable[i-1] . Green;
            NewPot -> Blue  = NewPot -> sBlue  = ColourTable[i-1] . Blue;
         }
         else
         {

            /* Now we have to combine two pots */

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

            Dist = 0x7fffffff; /* Largest distance between two colors */
            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;

                  /* If the new color is nearer than the old one, and at
                     least one is not locked
                  */

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

            /* Combine the two */

            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;
            }

            /* Really insert the new color */

            *NewPot2 = Pot[0];
         } /* else */

      } /* for i */
```

1.36 Picture/Dither

```
                Color adapts the
                active picture
                 to the current colors. The Floyd-Steinberg
algorithm is used for this.
```

1.37 Picture/Lock Colors

A window appears in which colors can be locked. Locked colors are ↩ not changed during Compute Colors . Only the custom method supports this, Median

Cut does not care about locked colors.

1.38 Picture/Change Colors

A color requester appears in which the current colors can be changed.

1.39 Picture/Load Colors

Using this menu item you can load colors from any IFF file. So if ↩ you want to adapt your pictures to Workbench’s colors, load the file ’ENV:Sys/Palette.ilbm’ and call Dither for every picture that will be shown on Workbench. To use colors from another screen, you will have to make a screenshot of it and then load this file.

1.40 Picture/Save Selected

This menu item allows you to save the active picture to disk. A filerequester appears in which you can choose where it should be written to.

1.41 Settings/SaveIcons?

Using this option you can save icons along with settings saved by ↩ Save As.
However these settings cannot be activated by double clicking on the icon.
Instead you must use

Project/Open
.

1.42 median

If this menu item is selected the Median Cut method is used for color

adaption
. This method does not support
locked colors.

1.43 custom

If this menu item is selected my custom method is used for color adaption.
It does support
locked colors
, but I don’t know if it’s fast or good.