VisualPrefs_eng

COLLABORATORS

	TITLE : VisualPrefs_eng		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		April 15, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

VisualPrefs_eng

1.1   VisualPrefs documentation

VisualPrefs 1.5c (31.10.99) by Massimo Tantignone

Overview

Requirements

The installation

The usage New!

Notes & hints ;-) New!

Is that all?

How to register

Translations New!

Thanks

Future plans

Known problems New!

Conclusion

Program history New! :-)

Mailing list New!

Author information

Items marked with "New!" have changed since the last major release,

not necessarily (although it's possible) since the previous update.

1.2   What is VisualPrefs?

WHAT IS VISUALPREFS?

====================

VisualPrefs is a patch which allows you to customize with great freedom

many aspects of the look of the Amiga GUI.

If you do want to see for yourself what VisualPrefs is able to do, just

take a look of the preview image supplied with this archive.

Using a comfortable preferences editor, you can configure many details like:

· different light/dark colors for active or inactive windows edges

· the thickness of all four window borders (also with system gadgets)

· the height of window/screen titlebars regardless of the font used

· the style of system gadgets (you can also use your custom bitmap images)

· the style of MUI/KingCON/ClassAct/DirOpus/Scalos titlebar gadgets

· the exact look of the BOOPSI and GadTools 3D frames

· the look of GadTools proportional gadgets

· the look of disabled GadTools button, cycle and string gadgets

· the position of window titles (centered in dragbar or in whole titlebar)

· the fill and text color of inactive window borders

· an optional "pop-up" behavior for GadTools cycle gadgets

... and also other interesting things. By the way, VisualPrefs contains all

the functionality (and more) of my previous patches Prop3D and CenterTitles.

You can configure individually the GUI of each different screen opened

by your applications, as well as define a global GUI look.

I know there are other programs doing some of this, but none of them has

all the features offered by VisualPrefs.

This program is shareware, but you can use it for a while without having

significant limitations . Also, the registration fee is very small ($6/10DM

if you want the keyfile by e-mail, $8/14DM if you want it by normal mail).

Please read carefully all of the documentation, as there are some things

about VisualPrefs that you need to know before using it.

Due to my studies, I can't promise anything about the future development

of VisualPrefs (except that there will be some). Probably it will proceed

very slowly. BUT, I will do anything in my power to fix any bugs you should

find in the current version as soon as possible.

However, keep in mind that I'm usually unable to do anything (except maybe

answer e-mail, but don't count too much on it) roughly from the end of July

to the middle of September, because of my summer vacation. :)

1.3   All the features of VisualPrefs

FULL LIST OF ALL THE FEATURES OF VISUALPREFS

==

o Twenty new pens for a more accurate GUI rendition. These pens can be

chosen independently for every screen, including the Workbench one.

o Possibility to define different shine/shadow pens for active and

inactive window borders. Never again those white and black edges! Now
you can have windows with plastic-looking borders, similar to the
X-Windows look (example: borders with a mid-blue background and
light/dark blue edges).

o Possibility to set the fill color of inactive window borders. This,
together with the previous feature, gives you complete control on the
window borders appearance.

o Possibility to set the color for inactive window titles.

o Ability to always keep square the screen and window titlebar gadgets.
Now, even with large fonts, you won't have those ugly tall & narrow
gadgets in the titlebar anymore.

o Ability to increase the screen and window titlebar height without
changing the screen font.

o Possibility to change width and height of the sizing window gadget, as
well as of the arrow gadgets. This allows to modify the thickness of
right and bottom window borders to finally have a correct aspect ratio
even with square-pixel (1:1) screenmodes. Window borders up to 32-pixel
thick! (Useful for very high resolutions, like 1600x1200...)

o Possibility to set the desired thickness for each of the four window
borders also when they don't contain system gadgets.
Now you can have, for instance, top and bottom window borders four-pixel
thick just like the left and right ones, rather than two-pixel thick as
usual.

o Many different styles for the frame of system gadgets (the ones in window
borders and screen titlebars).

o Four (for now) different styles for symbols drawn inside the system gadget
frames: Standard (but with fully scalable images), XEN (like SysIHack),
and two more, Future and Flat, with a simple, yet elegant design.
You can combine symbol styles with frame styles to achieve the most
personal look for your GUI.

o Additional ability to load and use custom bitmap images in place of the
internal system gadget symbols. You can replace just some symbols or all
of them, and also add a picture to the left side of the screen titlebar.

o You can choose to have "normal" scrollbars, scrollbars with a recessed
frame, scrollbars with a knob hole, or both (SysIHack-style). [Note: the
knob hole is only possible with BOOPSI propgadgets.]

o Possibility to freely configure the look of GadTools gadgets frames
(and, more generally, of all BOOPSI frames). Simple borders, double
borders, two-color borders, rounded borders, XEN borders, with any

background color, even a dithered one. Furthermore an optional new mode
to render "pressed" gadgets, more realistic than to simply switch the
border colors.

o Possibility to use custom bitmaps to replace the symbols of cycle and
checkbox GadTools gadgets.

o Possibility to also modify the look of the gadgets added to the titlebar
by MUI, KingCON, ClassAct, ViNCEd, DirOpus and Scalos.

o Compatibility with the KingCON scroller: forever gone the "overlapping
arrow gadgets" problem.
The same applies to SCMSG, SE and CPR.

o All the capabilities (and more) of my previous Prop3D patch are directly
integrated. You can configure every aspect of GadTools proportional
gadgets and arrow buttons!

o You can choose the look of GadTools cycle gadgets.

o GadTools cycle gadgets can have optional "pop-up menu" capabilities.

o GadTools string gadgets can have a filled frame (instead of edges-only).

o GadTools button and cycle gadgets can have a nice 3D look when disabled.

o GadTools string gadgets can have a special look when disabled.

o Window titles can be placed in the center of the dragbar or in the
center of the whole titlebar, just like with the CenterTitles utility. :-)

o Possibility to define the color used to draw the symbols contained in
system gadgets, or to fill their recessed areas if they have a 3D look.

o Possibility to have a full 3D screen titlebar by adding the left and top
bright sides.

o You can modify the preferences with a comfortable editor featuring a
preview zone similar to the one of the system Palette editor.

o Possibility to save default settings but also custom settings for
particular screens. Example: Wordworth, PageStream and TypeSmith don't
support window borders with a non-standard thickness? No problem, you
only have to change the settings for their screens.

o The preferences editor allows you to edit all system standard pens (like
backgroundpen, textpen, fillpen...) of the Workbench screen and to save
them in a way that is totally compatible with the Palette editor. In
fact, together with another program (FullPalette) which you can find
on Aminet, it becomes a complete replacement for said editor.

o For any screen it's possible to modify also the system standard pens
rather than just the VisualPrefs custom pens; this is optional, as with
some screens it could interfere with the handling of the GUI preferences
done by the programs which opened them.

In the future it is very likely that I'll add even more features; how soon
this will happen depends on how many people will request them, and on how
much time my studies will leave to me.

1.4   Limitations of the unregistered version

LIMITATIONS OF THE UNREGISTERED VERSION

=====================================

In the unregistered version there are no disabled features, but:

· You can only save/use up to 10 (ten) screen definitions;

· If you save more than 5 (five) screen definitions, a requester will appear

each time you run the GUI editor;

· Only two screens at a time can use the global template.

That's it. If you want unlimited screen definitions, unlimited template use

and no requesters, please consider registering ! :-)

1.5   Requirements for VisualPrefs

REQUIREMENTS FOR VISUALPREFS

===========================

An Amiga computer with AmigaOS 3.0+.

A display mode with square (1:1) pixels is strongly suggested, as well

as a screen resolution of 640×400 or more.

Also a graphic card could be useful (but not required).

1.6   How to install VisualPrefs

THE INSTALLATION

================

The installation is very simple; just use the supplied Installer script.

If you want to do a basic manual installation, follow these instructions...

First of all, copy the following files onto your hard disk:

+--+

| VisualPrefs To any directory (for example C:) |

| Prefs/GUI To SYS:Prefs |

| Prefs/GUI.info To SYS:Prefs |

| InstallFiles/select.gadget To SYS:Classes/Gadgets |

+--+

If you want, you can also copy the "GUISample.pre" file to SYS:Prefs/Presets,

and/or the sample bitmap directories to whatever place you like, but it isn't

really necessary.

Now you must add a line to your Startup-sequence (alas, it is not enough to

put it into User-startup), IMMEDIATELY BEFORE the one with the "C:IPrefs"

command: ——————————

Run > NIL: <dir>/VisualPrefs

Where <dir> is the name of the directory VisualPrefs resides in.

Example:

+-------------------------+

| ... |

| Run > NIL: C:VisualPrefs |

| C:IPrefs |

| ... |

+-------------------------+

Finally, if you were using the programs "Prop3D" and/or "CenterTitles",

remove them from your User-startup or WBStartup drawer.

1.7 How to use VisualPrefs

THE USAGE

=========

After the installation you must reboot. Then you can begin to use the GUI

editor to modify the interface's look.

I advise you to "play" a little with the editor at this point, before

coming back here to read the rest of the documentation.

In fact, the editor is rather intuitive, however there are some important

things to speak about that will be more understandable if you already know

the editor's basic structure.

Note: the term "DrawInfo pens" refers to the pens (colors) used for the

various elements of the GUI, such as light/dark edges, text, background, etc.

You find the list of DrawInfo pens at the left side of the editor window.

The GUI editor

Editing the GUI of single screens

Defining a default configuration New!

The pen mode

Color locking New!

Pop-up menus for cycle gadgets New!

Using external bitmap images New!

Special patches

Global options

Environment variables New!

Limitations of the unregistered version

1.8   The GUI editor

The GUI editor

With the editor you can set many aspects of the GUI, like the DrawInfo pens,

the style of window border gadgets, the width and height of the window

sizing/arrow gadgets and the exact look of gadget frames.

At the right side of the GUI editor's window there is a listview with several

options to configure the basic appearance of the gadget and window frames.

Above it is a preview display reflecting the current settings.

At the top left side of the editor's window there is a cycle gadget allowing

you to flip through various configuration pages for specific aspects of

the GUI look and behavior. These pages, also, can have sub-pages.

Above it is a text display gadget indicating the name of the current screen,

with a chooser gadget ("<") to go to the screen selection section .

Note that the effects of the various options in the listview at the right

side of the editor's window depend also on the colors you've chosen for the

screen's DrawInfo pens. If you choose the same color for more than one pen,

some options will seem to have no effect!

The cycle gadget below the palette has a special purpose that will be

explained in the section " The pen mode ".

1.9   Editing the GUI of single screens

Editing the GUI of single screens

You can edit the parameters of single screens. You can select the screens

to edit by clicking the "<" gadget in the top left of the window. Then you

can double-click on a screen name from those in the listview.

For a screen to be edited the first time, it must of course be already open,

otherwise it won't appear in the listview.

Any subsequent time the editor will open a clone of the screen if the "real"

one is not open.

Names of screens not already saved in the preferences file will appear in

a "highlight" color.

You can also inhibit VisualPrefs from operating on certain screens, by

clicking on the cycle gadget ("Patch"/"Ignore") at the bottom right of the

window. If you select "Ignore", the name of the screen will appear recessed

and VisualPrefs won't patch it at all.

VisualPrefs recognizes screens by comparing their public name (or title if
they aren't public) with the name saved with the GUI editor.
You can modify this name with the string gadget; in particular, to make it
match a larger class of screens, you can add a #? at the end.
This is NOT true pattern matching, however: wildcards don't work inside
the name, only at the end (and only the "#?" couple).
You can delete a screen definition using the "Delete" gadget, but only
if the GUI editor is not present on the corresponding screen. Also you
can't delete the definition for the Workbench screen. If you attempt to do
so, an error message will be displayed.
If the screen associated to the definition you delete is still open, its
name will remain in the list, but it will return to a "highlight" color,
which indicates its definition won't be saved into the preferences file.
For any change to take effect, you must then click on "Use" or "Save" on
the main editor panel.
Note that the GUI editor remembers the settings of all "visited" screens,
so you'll need to select "Use" or "Save" only once at the end of your
editing session.
VisualPrefs will attempt to close and re-open the Workbench screen each time
you change something significant in its configuration; screens of other
applications will have to be manually closed and re-opened for all changes
to be correctly applied.

1.10   Defining a default configuration

Defining a default configuration

You can define a default configuration ("template") which will be used for
all screens lacking an entry in the preferences file. To do this, just edit
the parameters of any screen to reflect your desired default settings, then
select "Save as template" (or "Use as template" for a temporary change) from
the "Project" menu.
You'll be asked some questions Changed! about how the DrawInfo pens will
have to be chosen by VisualPrefs for the various screens, then your default
template will be saved.
DON'T click on "Save" or "Use" now, but simply quit the editor, or else the
default template settings will also be saved as the current screen's settings!
New! If you later want to modify your saved template, all you have to do is
to select the "Recall Template" menu item, do your changes, and save the

template again.

New! Note that, depending on which screen you're editing when you recall
the template, the GUI editor might be put in a "special mode" where the pen
colors shown in the preview area (and in the pen listview) don't necessarily
correspond to the pen index numbers which will be saved with the template;
rather, the preview colors are chosen to best represent your template
settings, but the previously saved pen index numbers are preserved unless
you change them by hand using the palette gadget (until you do that, the
color box in the pen listview will have an "X" superimposed to it).

1.11   Save options for the global template

Save options for the global template

When you save the configuration for the global template, a requester will
appear asking you some questions about the method to be used for the choice
of DrawInfo pens on different screens.

This is necessary as each screen has its own particular palette, so it isn't
convenient to always use the same "fixed" pen settings for all screens.

In the following text, "current screen" means "the screen on which there is
currently the GUI editor window", whereas "destination screen" means "the
screen to which the template will be applied".

The first thing to decide is if you want to save the current pen settings
in order for them to be used by the template. A set of radio buttons offers
two choices:

"Use current settings" means the current screen's pen settings will be saved
along with the template; this way, the destination screen's pens will be
chosen in a way that respects those settings as much as possible.

"Choose pens automatically" means the destination screen's pens will be
assigned some suitable values which only depend on the already existing
(system standard) DrawInfo pens of the screen itself.

This guarantees an acceptable look, although not always the best possible
one; anyway, if in doubt, you should probably choose this option.

If you click on "Use current settings" another set of radio buttons becomes
available (otherwise it is disabled), which you can use to tell VisualPrefs
how the "bad" pens will have to be "reassigned". "Bad" pens are pens of the
destination screen which, if they were to keep the same index, would have a
visually different color than the one they have on the current screen.

You have four possible choices:

"Use fallback values" assigns pens the same way "Choose pens automatically" would, but it only affects the "bad" pens (the pens which already have a correct color will be left alone).

"Use selected values" means the destination screen's pens will use exactly the same color index values as the current screen's pens, even if the actual colors don't look the same.

"Find best match" means VisualPrefs will try to assign the pens those colors which most resemble the ones assigned to the current screen's pens, even if they don't have the same index values.

"Allocate new color" is similar to "Find best match", but also creates a new color which is identical to the wanted one if no already existing color is similar enough. This new color is not locked by default.

Then there is a slider, "Bitplane threshold", which you can use to tell VisualPrefs what is the minimum screen depth under which it must assign simple default values to the GUI pens (as with the "Choose pens automatically" option). This allows you to perform pen color remapping on screens with a large palette, yet obtain accettable results on screens with a small number of colors (where color remapping usually gives poor results).

Finally, there are three checkboxes offering additional options:

"All pens" is used to indicate whether the template must affects all the GUI pens of the destination screen, or rather just the custom pens added by VisualPrefs.

"Lock pens" is used to tell VisualPrefs to lock all the GUI pens on the destination screen, regardless of how they were determined. This is available only if the "Use current settings" option is chosen, as otherwise the GUI pens get locked automatically anyway (being the original DrawInfo pens).

"Allow nameless screens" is used to tell VisualPrefs to apply the template even to screens having no title (which might sometimes cause trouble, so take care with this option).

1.12   The pen mode

The pen mode

As you already know, each DrawInfo pen has a color index associated to it. VisualPrefs, with its GUI editor, allows you the greatest flexibility in the selection of colors for the DrawInfo pens of a screen.

Why? Some screens, when they change their depth, remap the colors of their palette in a different way than others (the Workbench screen is a perfect

example of this, as it always remaps four of its eight colors to the end

of its palette).

So it's not always sufficient to simply assign a fixed color index to a pen.

For this reason, each pen has a "pen mode" that determines how its color

index is recalculated when the screen's depth (i.e. number of colors)

is different than what it was the LAST TIME you used the GUI editor on it

to edit (and save) the pen's color.

To set the pen mode of a pen you use the cycle gadget ("Absolute"/"Relative")

located below the palette in the "Pens" page of the GUI editor.

Let's see how this works; we'll indicate with "n" the generic number

of colors of a screen.

1) If the pen mode is "Absolute", the pen will always use exactly the

absolute index of the color AS IT WAS WHEN YOU SELECTED IT; this

can go from 0 to n - 1.

If, later, this color index doesn't exist on the current screen because

there are too few colors, the result is undefined (most likely the

wrong color will be used).

2) If the pen mode is "Relative", there are two possibilities:

2a) If the color index, when you selected it, was in the FIRST HALF

of the palette (color 0 to n/2 - 1), then it will be treated as

in the "Absolute" case, EXCEPT when there are so few colors that,

maintaining its absolute index, it would end up in the second half

of the palette (or even beyond). In this case, a fallback color

index will be used.

2b) If the color index, when you selected it, was in the SECOND HALF

of the palette (color n/2 to n - 1), then it will always keep the

same offset, i.e. distance, from the END of the palette rather

than from the beginning. So, for instance, color n - 3 will always

be color n - 3 regardless from how many colors the screen has.

Furthermore, if the screen has so few colors that, maintaining its

offset from the end, the color would end up in the first half of

the palette (or even beyond), a fallback color index will be used.

In the case of the Workbench screen, for example, you should set "Absolute"

for pens using the first four colors and "Relative" for pens using the last

four.

What you should do for pens using other colors in-between depends on many

factors, one of which is whether or not you use some utility allowing you

to set and lock all of the Workbench colors (rather than just eight of them

like the standard Palette editor does).

1.13   Color locking

Color locking

As some programs change their palette while running (for example web browsers)
you could need to lock the colors you assign to VisualPrefs-specific DrawInfo
pens. In the "Color locking" page of the GUI editor you can do just that.
There you find ONLY colors that VisualPrefs will use and that will NOT be
locked automatically anyway by Intuition.
Note that this still doesn't guarantee to get the intended color for a pen:
if you use color locking, you must also make sure the color you chose already
has the desired value when the screen is opened! All VisualPrefs does is to
prevent such value from changing later.
New! If the "All pens" option is turned on for the screen you're editing,
an additional "Auto lock std" checkbox is available in the "Color locking"
page; this checkbox is used to tell VisualPrefs if the system standard
DrawInfo pen of the screen are to be always automatically locked (like
Intuition would do) or rather they are to be left initially free, and then
selectively locked according to your color locking preferences (which, for
"All pens" screens, can also include colors used by standard pens).
If "All pens" is off, the standard pens will be always locked.

1.14   Pop-up menus for cycle gadgets

Pop-up menus for cycle gadgets

Although this is more of a functional patch than an aesthetic one, starting
from release 1.4 VisualPrefs can add pop-up capabilities to GadTools cycle
gadgets.
To this purpose, an external BOOPSI class library called "select.gadget" is
needed; it provides a class of pop-up gadgets and other useful functions
used by VisualPrefs to implement the pop-up patch for normal cycle gadgets.
Such library is included in the VisualPrefs distribution archive, and gets
installed by the installation script if you tell it to do so. However, the
"select.gadget" library is also a stand-alone product aimed at developers,
and is therefore available on Aminet in a separate archive, with full
documentation and support files (dev/gui/select_gc.lha).
What this means is that at any given time the stand-alone release of
"select.gadget" will be the most recent available; the one contained in the

VisualPrefs archive will always be either the same one or an older one
(unless this is a beta release of VisualPrefs, in which case the library
might be a beta version as well).
In the latter case, you just have to download the stand-alone release and
overwrite your old copy of the library with the newer one.
If you don't plan to use the pop-up patch for cycle gadgets, you can avoid
installing the library at all. Of course, you can always install it later
by hand (it should reside in SYS:Classes/Gadgets).
To activate the pop-up patch (once the library is installed) it is sufficient
to move the "Delay" slider in the GadTools/Cycle page of the GUI editor.
When the slider is completely at the right, the patch is turned off. Any
other position tells VisualPrefs how much time (in 100ths of a second) to
wait before displaying the pop-up menu when a cycle gadget is pressed.
If you click on the "symbol" zone of the cycle gadget, however, the pop-up
menu won't appear regardless of the delay value.
You can set the colors used to highlight the active item in the pop-up menu
by changing the Select, Select Text, Select Shine and Select Shadow pens.
New! The pop-up menu panel can be displayed using three different methods:
the default is using a normal Intuition window, which is somewhat slow but
gives the greatest compatibility in all circumstances.
You can also choose to draw the panel directly on the screen's bitmap, which
allows for smoother operations, but might cause some display corruption if
a program is drawing something on an overlapping window.
The third method also renders directly on the bitmap, but locks the screen's
layers in order to prevent display corruption. If you use this method, you
should be very careful not to use commodities or other input handlers which
might try to lock the layers as well while the panel is being displayed, or
you risk a deadlock.
There are some other options to configure the look and behavior of the pop-up
menu (position, highlight style, height of menu items... even a "drop shadow"
option); just play a little and experiment with the various combinations to
find out what works best for you. Alas, currently there's no preview available
for the pop-up menu, therefore after any changes you have to select "Use"
and then restart the GUI editor (or another program having a GUI with cycle
gadgets) to try out the results. In the future I'll try to add a test gadget
to let you try your changes immediately.
As a final note, if you use the pop-up feature of VisualPrefs you should turn
off any other similar option such as MCP's "CycleToMenu".

1.15   Using external bitmap images

Using external bitmap images

Starting with release 1.5, VisualPrefs allows to use custom-made bitmap

images for system gadgets. All relevant settings for this feature are

editable in the System page of the GUI editor.

System gadget images are composed of two parts: the image button frame (which

you can choose with the "Frame" cycle gadget) and the actual symbol contained

within it, which can be a vector or bitmap image.

You can select a style of vector symbols with the "Gadget style" listview.

With the "?" button immediately below it, you bring up a file requester which

allows you to select a style of bitmap images residing on your hard disk.

Bitmap image styles are organized in separate directories, named after the

style itself, which contain a number of sub-directories. Each sub-directory,

in turn, is named after a particular symbol and contains the actual bitmap

image files (brushes) for that symbol.

The image files can have any format supported by your picture datatypes.

To remove the currently selected bitmap style, just enter an empty string

in the file requester.

In the VisualPrefs distribution archive you can find one or more (depending

on the version) directories of sample bitmap styles; you can use them to

experiment with using this feature before starting drawing your own images.

When the file requester appears, simply select the name of one of these

bitmap directories (for instance "Sample-8c").

Currently, VisualPrefs uses two image files for each symbol: a file called

"Default" containing the images for all possible states of the symbol

(inactive, normal, selected, etc.) and an optional file called "Default.mask"

which is an one-bitplane image used to mask out certain pixels of the symbol

image when it is blitted onto its button frame.

In the main file all the images for the states must be placed horizontally,

next one another, in a specific order (see below). All the state images must

have the same width.

The mask image, of course, must have exactly the same size as the main one.

A symbol state image is always placed by VisualPrefs in the center of its

surrounding button frame; if the symbol image is too big for the button

frame, only its center part will be shown.

If the main image file for a particular symbol can't be loaded, the vector

symbol from the currently selected vector style will be used in its place.

This means you can also "patch" an already-existing vector style, by drawing

just a few bitmaps to replace only those symbols you don't like.

In the most extreme case, you could use this feature to simply add a logo

to your screen titlebar, and keep using a vector style for everything else!

As already mentioned, each symbol image file must contain a number of images

corresponding to the various states the symbol can have. The following list

shows the names of all currently supported symbols along with the number of

states they require and the exact left-to-right order the states must have.

Note: "Normal" means "not selected and not inactive".

Directory | Explanation | # | Left-to-right order of states

-----------+----------------+---+---

SDepth | Screen depth | 2 | Normal, selected

Close | | 3 | Normal, selected, inactive

Zoom | | 3 | Normal, selected, inactive

Depth | | 3 | Normal, selected, inactive

Iconify | | 3 | Normal, selected, inactive

Snapshot | | 3 | Normal, selected, inactive

MUI | | 3 | Normal, selected, inactive

PopUp | | 3 | Normal, selected, inactive

Padlock | DOpus "lock" | 4 | Normal, selected, inactive, inact. selected

Size | Sizing gadget | 3 | Normal, selected, inactive

ArrowUp | | 3 | Normal, selected, inactive

ArrowDown | | 3 | Normal, selected, inactive

ArrowLeft | | 3 | Normal, selected, inactive

ArrowRight | | 3 | Normal, selected, inactive

SBarLogo | Screen bar logo | 1 | Normal

Checkmark | | 4 | Normal, selected, disabled, dis. selected

Cycle | | 4 | Normal, selected, disabled, dis. selected

Fallback | Unused for now | 3 | Normal, selected, inactive

Please give a look to the "Sample.ilbm" picture, found in this archive within

the "Prefs/Bitmaps" directory, to see a graphic representation of the above.

In the future it is possible that VisualPrefs will support different image

sizes for each symbol, or even bitmap scaling of the "Default" symbols.

1.16 Special patches

Special patches

In the "Global" menu you find some entries that allow you to selectively

activate some patches. They can add some overhead to the working of

VisualPrefs, so you have a choice whether to use them or not.

Select "Patch KingCON" to tell VisualPrefs to patch the gadgets added by

KingCON to window borders.

Select "Patch AmigaGuide" to tell VisualPrefs to correctly display the

text in the toolbar gadgets of an AmigaGuide document (without the patch

it is written in JAM2 mode, covering part of the gadget frame).

Select "Patch SAS/C" to tell VisualPrefs to patch the gadgets added

by SCMSG, SE and CPR to window borders.

Select "Patch MagicMenu" to tell VisualPrefs to center (vertically)

in the screen bar the menu titles displayed by MagicMenu. This is useful

if you have increased the screen bar height.

Use this patch with MagicMenu 2.19+ only! With earlier versions it just

makes things worse... ;)

1.17 Global options

Global options

In the "Global" menu you find some entries that allow you to change globally

(that is, for all screens) some aspects of the VisualPrefs behavior.

Select "Old XEN look" to tell VisualPrefs to use a XEN style for system

gadgets that is more similar to the one of the original SysIHack patch

(which first introduced it); otherwise VisualPrefs will use a XEN style

that is slightly different from SysIHack's one.

For now you will only notice a difference in the sizing gadget.

Select "Don't use Template" to tell VisualPrefs NOT to patch any screen

which doesn't have a definition in the preferences file; that is, if this

option is selected, only saved screens will be patched.

Select "Bound hit box" to tell VisualPrefs that system gadgets should only

react to clicks made within their visible "button" part. If this

option is turned off, the gadgets react to any click made on their hit box,

even if visually it is outside the "button" image.

1.18 Environment variables

Environment variables

Some very specific aspects of the behavior of VisualPrefs can be configured

by the use of environment variables which must present at boot in the

ENV:VisualPrefs directory. The currently recognized variables are:

WBOpenDelay

Set this to the number of 50ths of second VisualPrefs must wait before

reopening the Workbench screen when it has been closed by the program

to update its GUI. Increasing this number can help to avoid occasional

lockups which can happen due to several reasons (IPrefs, RTG software,

other patches). If you never experience lockups at the opening of the

Workbench screen you don't need to use this variable.

BirdieMode

This is useful only if you are using the Birdie patch with the NOBITMAPS

switch on. Set this variable to NOBITMAPS to let VisualPrefs know it

doesn't need to disable all of its optimizations even if the "Don't

optimize border rendering" option is active (which is needed to achieve

compatibility with Birdie). If you aren't using Birdie, or if you're

using it without the NOBITMAPS switch, you shouldn't use this variable.

CopyGTText New!

Set this to 1 if you want VisualPrefs to copy the label text of GadTools

button gadgets when applying its patches to them. This fixes problems

which can happen with software that incorrectly reuses the same string

pointer when creating a list of GadTools gadgets. The official Amiga

developer documentation explicitly states this is illegal because the

gadtools.library doesn't copy the strings but rather it just references

them; however, some programs do it anyway as up to AmigaOS 3.1 the

gadtools.library actually does copy the strings. Go figure.

SafetyLevel New!

This is an integer value in the range 0..2 which you can set to tell

VisualPrefs how far it has to go in its attempts to avoid a stack

overflow within its patches to system BOOPSI classes.

0 - No stack checking (the default). This is enough in most cases,

especially if you're running some stack extension utility like

StackAttack. However, StackAttack can't prevent stack overflows

in the input.device context; if you experience such a problem,

it's advised you choose an higher level setting.

1 - If the available stack size is lower than a minimum amount, the

VisualPrefs custom code isn't executed; instead, the original

system code is used. If the available stack size is MUCH lower

than the minimum amount, not even the system code is executed.

Level 1 avoids stack overflow crashes, but can cause some unexpected

side effects, like corrupted graphics or unusable gadgets. It is

therefore recommended to set the minimum stack size to a low value

in order to only trigger this safety measure when it's really needed.

2 - If the available stack size is lower than a minimum amount, the stack

is replaced by a new, larger one. The old stack is put back into place

when the VisualPrefs custom code returns. This is performed in a very

optimized way, avoiding memory allocation when it isn't necessary.

Level 2 is the recommended setting if you experience frequent stack

overflow crashes using VisualPrefs.

MinStackSize New!

This is an integer value in the range 300..4000 which you can set to

define the minimum stack size (in bytes) below which a safety measure

is triggered in BOOPSI classes patches (if "SafetyLevel" is greater

than zero). The default is 1000 bytes.

Special New!

This variable is a flags longword. It contains a number in decimal

format and each bit of the number is like an on/off switch for some

option. You can OR:

1 - To change slightly the look of the "pop-up" symbol for system gadgets

in the XEN style.

2 - To have the screen depth gadget always as wide as the window depth

gadget, even if the screen titlebar has a different height than the

window titlebar.

4 - To have XEN button frames with thicker (2 pixels) inner edges.

8 - To make VisualPrefs use FAST memory rather than CHIP memory, if

your RTG software offers this feature (for instance, if you're using

CyberGraphX and have the PLANES2FAST variable set to "1").

16 - To prevent VisualPrefs from adjusting the results of the IM_FRAMEBOX

method of "frameiclass".

32 - To make VisualPrefs use its old (less optimized) semaphore system

rather than the new one as introduced in release 1.5a. The new system

is in general more stable, but some people with specific setups might

still find the old one working better for them. Don't use this if not

really needed!

All other bits are reserved for now, just set them to zero.

Example: SetEnv VisualPrefs/Special 12 enable thick XEN frames + use FAST

1.19   Important notes and useful hints

NOTES AND HINTS

===============

The most important note of all: VisualPrefs works well with all programs

respecting the rules, like adapting to the window border sizes, using

system functions for creating/drawing gadgets, etc. Of course, badly written

programs or programs using proprietary code for their interface (like MUI

applications) will benefit very little from VisualPrefs.

——

When using VisualPrefs you should disable all other similar GUI patches,

such as SysIHack, UrouHack, MagicFrames, CycleToMenu and MCP's FrameIHack,

SysIHack, PropHack, CycleToMenu and NewGadTools options (but for the last

one see also the next note).

Also, you should remove my previous Prop3D and CenterTitles utilities.

——

NOTE WELL: If you use the "WBGauge" option of MCP, make sure that the

"NewGadTools" option is ENABLED as well, or else there could be crashes

(at least with some versions of MCP).

You should be able to turn on "NewGadTools" without any conflict with

VisualPrefs, as long as you leave deactivated all individual patches.

——

VisualPrefs, like most patches, causes a slightly larger stack usage by

programs calling the functions it modifies. Normally this isn't a problem,

but it is known that the IPrefs program allocates a very small stack for

itself and sometimes it could run out of it (for example when using certain

datatypes for the Workbench background patterns). If you experience this

problem, I advise you to use FastIPrefs 40.35 (util/boot/FastIPrefs4035.lha)

as a full replacement of the original IPrefs.

New! Note: do not use FastIPrefs 40.36b, it has some serious bugs!

Another program that suffers from a small default stack is WShell; to solve

this, it suffices to increase the stack size in the DHOpts command using its

STACK parameter (DHOpts CON:/xx/yy/ww/hh STACK 8000).

——

A good way to ensure you have enough stack in most cases is to use the

StackAttack patch by Georg Steger (Aminet: util/boot/StackAttack.lha).

Thanks for this useful program Georg!

——

VisualPrefs can be made nearly 100% compatible with the Birdie patch.

This is not automatic, as it requires disabling some optimizations

in the code, and this wouldn't make sense if you aren't using Birdie.

In order to obtain compatibility with Birdie, you must make sure that:

· The option "Don't optimize border rendering" is turned on for any screen

you want to use Birdie on.

· The Background pen and the Inactive Fill pen are both set to color zero for

any screen you want to use Birdie on.

Also, if you use Birdie with the NOBITMAPS switch, you should create an

environment variable called "VisualPrefs/BirdieMode" containing the string

NOBITMAPS. The variable must exist when VisualPrefs is started. As of

VisualPrefs 1.4, if you don't use the switch you don't need this variable.

Furthermore, note that on 15/16/24-bit screens, the Glyph pen will be rendered

as a darker shade of the window border pattern, unless you set it to be

the same as the Shine pen (in which case it will be a lighter shade of

the border pattern).

All this was tested with Birdie 1.5; I can't guarantee it will continue to

work with future versions of the patch.

——

New! It seems that in some cases MWB-Demon can cause problems if VisualPrefs

and Birdie are both running. If this is your case, you can replace MWB-Demon

with another utility which sets and locks the needed Workbench colors.

——

There's a bug in IPrefs 3.0/3.1 which can cause a deadlock when

VisualPrefs reopens the Workbench screen after a change to its settings.

This problem shouldn't happen very often, but if you experience it, you have

two ways to try avoiding it:

· By telling VisualPrefs to wait a little before reopening the screen. Just

set the environment variable "VisualPrefs/WBOpenDelay" to a number

indicating the duration of the delay.

A value of 50 means a second, 100 is two seconds, and so on; normally it

should suffice 20 or less.

· By installing the PatchWB program by Thomas Richter, supplied in the

VisualPrefs archive. The installation script will install it if you tell

it to do so. This should fix the problem once and for all.

New! Don't do any of the above if you don't have the problem!

New! The problem does not exist with IPrefs V44 from AmigaOS 3.5.

——

If you use NewIcons, make sure it is launched after VisualPrefs

at the boot, or else its "No borders" function won't work properly.

New! Put it after (Fast)IPrefs too, or similar problems could arise.

——

For better results with VisualPrefs, use MagicMenu 2.19 or greater;

versions up to 2.15 display the menu bar incorrectly if the titlebar height

has been increased.

——

To patch the "iconify" gadget of ClassAct programs, you must select

the "Normal" border gadget in the ClassAct preferences editor; if you select

"3D" the patch won't work.

——

VisualPrefs works best with 1:1 resolutions (i.e. with square pixels).

With other resolutions it works, but the results aren't always optimal.

Furthermore, in low-res (320×???) it still has some problems (better to

set such screens to "Ignore" in the screen list).

Also, keep in mind that the GUI editor needs some space for its window;

on small screens its user interface may not fit entirely.

If this happens, try reducing the size of its font with the FONTNAME and

FONTSIZE tool types.

——

VisualPrefs greatly benefits from an utility allowing you to define and

lock ALL Workbench colors, as opposed to just eight of them like you can

do with the standard Palette editor.

There are several such utilities on Aminet, and there is also one written

by me, "FullPalette", available on Aminet as util/wb/FullPalette22.lha

(as I write this; the trailing number, 22, could have changed by now).

Of course I suggest you use this one. :-)

New! If you use FullPalette, make sure you have at least version 40.20 of

FPPrefs, or else the bitmap loading feature of VisualPrefs won't work

properly. You can find an adequate version of FPPrefs in all the archives

of VisualPrefs 1.5.

1.20 Is that all?

IS THAT ALL?

============

No. I certainly forgot to mention something, so if you have any problem with

the usage of VisualPrefs and/or the GUI editor, don't hesitate to ask me for

clarifications.

1.21   So you want to be a registered user?

HOW TO REGISTER TO VISUALPREFS

==============================

Please read carefully the following text!

The method of registration is very simple. You only have to print the

registration form , fill it in and send it to me along

with the required sum of money.

The registration fee is:

· $6 (10 DM, or equivalent) for shipment of the keyfile by e-mail, or

· $8 (14 DM, or equivalent) for shipment of the keyfile by normal mail.

For now I allow you to pay me in any currency, although in the future this

could change.

Of course, if you pay me in italian Lire I will be happier... :)

You can choose between various methods of payment:

o Send money in an envelope, together with the registration form

o Make an international postal money order, and send the registration

form in an envelope (specify "VisualPrefs registration" on the money

order form)

o Make a bank draft, and send it in the envelope together with the

registration form (please, first verify that your bank draft can

be cashed in Italy)

Sorry, at the moment I can't accept credit cards.

If you really want to, you can also send me the registration form via e-mail,

to make sure I receive your order, however this is mostly useless. But PLEASE

put it ALSO in the envelope with the money or bank draft.

You can choose whether to receive the keyfile via e-mail or via normal mail.

o If you decide to receive it via e-mail, please specify your e-mail address

on the registration form.

o If you decide to receive it via normal mail, please enclose a pre-formatted

floppy disk (DD or HD) into the envelope.

I will ship you the keyfile as soon as I have received BOTH the registration

form AND the money. If you request a shipment via normal mail, please allow

2-8 weeks for delivery.

IMPORTANT NOTE: as stated elsewhere in this guide, I'm usually unable to do

anything about VisualPrefs for a significant part of summertime, due to my

vacation. This also includes shipping keyfiles! So don't worry if you don't

receive anything before mid-September; when I'm back, I'll immediately

catch up with all the registration requests I'll have received.

I'll repeat my address here:

Massimo Tantignone

Via Campagnoli, 4

28100 Novara (NO)

ITALY

If anything is not clear, please write to me for more information. Thank you!

1.22   VisualPrefs registration form

VISUALPREFS REGISTRATION FORM

============================

Please print and fill in this registration form, then send it together

with the payment to:

Massimo Tantignone

Via Campagnoli, 4

28100 Novara (NO)

ITALY

8<-----------------------------CUT HERE---------------------------------

==================== VisualPrefs registration form ====================

====================== valid for all countries ======================

---- Please mark with an 'X' the boxes relevant to your choices ----

----

I would like to become a registered user of VisualPrefs. To this purpose,

along with this registration form, I'm sending (enclosed in the same

envelope, or separately if this form is sent via e-mail):

[ ] The required sum of money (see below) as cash

[ ] A bank draft for the required sum of money

[ ] A document certifying the payment of the required sum (example: receipt)

[ ] Nothing, because I have sent/will send a postal money order

I acknowledge that the required sum of money for the registration is:

· $6 (10 DM, or equivalent) for shipment of the keyfile by e-mail, or

· $8 (14 DM, or equivalent) for shipment of the keyfile by normal mail.

So, I want to receive my personal keyfile, as soon as possible, by:

[ ] Electronic mail (e-mail) at the address specified below

[ ] Normal mail (please enclose a pre-formatted floppy disk)

If the keyfile is sent via normal mail, I allow 2-8 weeks for delivery,

starting from the day the author will have received both the registration

form and the money.

My personal information is:

First name _____ Last name _____

Street _____ Number _____

City _____ State _____ ZIP _____

Country _____

E-mail address _____@_____

VisualPrefs version I'm using _____

(Optional) Configuration of the Amiga(s) I own _____

_____

I guarantee that the keyfile I will receive won't be distributed and

will be used only by me on one or more computers owned by me.

Date _____ Signature _____

1.23   Translations to other languages

TRANSLATIONS

============

If you need a version of the catalog and documentation in your language, try

looking on Aminet for a file named "util/wb/VisualPrefs_XX.lha", where in

place of "XX" you should put the Internet domain suffix of your country

in capital letters (example: "util/wb/VisualPrefs_DE.lha" for german users,

"util/wb/VisualPrefs_RU.lha for russian users, and so on).

If you don't find it, and you want to write it yourself, please e-mail me to

ask if someone is already doing it. If nobody is, you will be assigned the

job :)

If you DO find it, it might be that its contents refer to an earlier release

of VisualPrefs. In that case, just be patient and wait until the translators

release the new version...

Thanks to ATO and my other independent translators for the excellent work!

1.24   Thanks to...

THANKS TO...

============

Many thanks go to my faithful (well, most of them anyway :) beta-testers,

without which the development of VisualPrefs would not have been possible.

They are, strictly in alphabetical order:

Alex Pagnoni

1.25 Future plans for VisualPrefs

FUTURE PLANS FOR VISUALPREFS

===========================

Well... of course I would like to add more image styles, enhance the ability

to use custom bitmap images for system gadgets, add new windows and gadgets

capabilities, and so on. Also, an interface for developers, probably through

a freely distributable "visualprefs.library", would be an interesting idea.

Many users did send me their ideas for enhancements. Thanks to all of you!

Although I can't promise anything, your suggestions will be seriously

considered in future developments.

Some people wrote me that the GUI editor sucks, and I tend to agree.

It was developed alongside VisualPrefs, so a good design wasn't so much a

priority as the support of all the VisualPrefs features. Now it should really

be rewritten from scratch, and maybe someday I'll find the time to do it, too.

Certainly I'll have to make VisualPrefs much more modular before adding

any major new feature. Otherwise, the executable would become too big (it

is already).

New! Luckily this wasn't necessary to support custom images for system

gadgets, after all ;-)

My idea is that each style for the various GUI elements should become an

external module (like a shared library), so that the user can choose

whether to install it or not; this way it would be possible for anyone to

create his own custom styles and also to distribute them. Indeed, this is

what I'm mainly working on at the moment...

How much of this will be done depends on several issues, in particular:

1. How much time my studies will allow me to spend into Amiga programming;

2. Which features the future AmigaOS 3.5 GUI will already have;

3. What kind of feedback I will receive from Amiga users.

So, if you want to see new features added to VisualPrefs, please register! ;)

1.26   Known bugs and problems of VisualPrefs

KNOWN BUGS AND PROBLEMS OF VISUALPREFS

======================================

Ahem. Here you'll find some information about the problems you may currently

encounter using VisualPrefs.

o New! Some users of Phase 5 PPC cards reported starting having strange

problems with VisualPrefs after they updated their card's flashrom to

the most recent version available on P5's FTP site. This seems to happen

especially to people having also a CyberVisionPPC or BVisionPPC graphic

card. Strangely enough, VisualPrefs works normally on the same systems

without the PPC card or with an older flashrom version.

I don't know yet what the problem could be; of course I'm seriously

investigating on this, but my lack of a PPC card definitely doesn't help

the matter. It appears to be a stack overflow problem in most cases, but

this doesn't seem to cover every single fact yet...

o New! Apparently, on some systems using old versions of FBlit together

with VisualPrefs can corrupt the appearance of system gadgets. According

to Stephen Brookes (FBlit's author) this should have been fixed in the

latest versions of the program, available from http://www.tpec.u-net.com.

o Certain windows of some MUI programs (like IBrowse) appear empty if the

top border's thickness is different than (titlebar height increment + 2).

This is really a nasty problem, I don't know if I will ever be able to

find a solution.

For now, set the top border to a value == (titlebar height increment + 2).

o MagicMenu 2.19+ will always display the menu bar with the correct height,

but the menu titles aren't centered vertically within it. So, for now it

is necessary to use the "Patch MagicMenu" option of VisualPrefs to obtain

the centering.

I've already explained the problem to the author of MagicMenu and hope

he'll address it soon.

o Prop3D seemed to cause problems when used together with ARQ _AND_

MagicMenu, but I haven't been able to reproduce this bug yet. Now let

me know if this still happens with VisualPrefs.

o There are still some quirks in the GUI editor which I've been thus far

unable to trace. Keep those bug reports coming... ;)

o Display problems using Thor on the Workbench screen. It turned out this

isn't due to VisualPrefs per se, but only to the choice of colors for your

shine/shadow/fill pens. If you have this problem, try to select pens that

are all as "near" as possible in the palette.

In particular, it should work if all of the pen indices give the same

(integer) result when divided by 4 (example: pens 4, 5, 6 and 7).

1.27 Conclusion

CONCLUSION

=========

When you'll encounter a bug/problem/whatsoever, report it to me writing as

much details as possible on what happened and on your system configuration

at the time, including any patch or commodity running in background.

It's possible I already know about some bugs, but that they can't be fixed

due to current limits of Intuition.

Please write also to say VisualPrefs works well on your systems, if it does!

You can also write suggestions for future enhancements, but keep in mind that

I must always give absolute precedence to the elimination of bugs.

In fact, I'm not even sure to which extent it will make sense to continue

the development of this patch, because, whatever will be the successor of

the current AmigaOS, it probably won't need VisualPrefs anyway (I hope!).

So, for now I think it's more important to remove bugs and problems of the

current version.

If you are an english speaker, please check the correctness of this

documentation and report to me all grammatical or syntactic errors. ;-)

In the end, remember that VisualPrefs is an "ongoing" project and as such it

can always undergo changes of plans (alas, sometimes they don't depend on

my will).

I apologize for not being always able to answer your mails immediately,

but my studies don't allow me to use my Amiga as often as I would like to.

However, I do my best to answer ALL e-mail eventually.

That said, have fun with VisualPrefs!

Massimo Tantignone (tanti@intercom.it)

[http://www.intercom.it/~amigaws]

1.28   The history of VisualPrefs

THE HISTORY OF VISUALPREFS

=========================

1.5c (31.10.99)

Bug-fixes:

· Now the screen depth gadget on screens ignored by VisualPrefs works again

even with the "Bound hit box" option turned on.

· When using a custom bitmap for DirOpus "padlock" images, the "active

selected" image was used even for the "inactive selected" state. Now fixed.

· Fixed a problem which prevented custom bitmaps to work if the Picasso96 or

OS 3.5 picture.datatype was used on CyberGraphX systems. A nice side-effect

of this is that now VisualPrefs can always load bitmaps in V43 mode and

therefore display truecolor images as they are meant to be, if the screen

is deep enough.

· Now titlebar gadgets in the preview of the OS 3.5 Font preferences editor

are drawn with the correct height.

· Fixed a problem which caused garbage or black rectangles to appear in

window borders on deep screens if VisualPrefs' "Don't optimize..." option

was turned on, a "GadTools" or "Filled" frame style was used for system

gadgets, and PatchControl was started before VisualPrefs in the

Startup-sequence.

· Fixed a bug which caused system gadget images to look trashed if the

"RTG mode" bit was set in the "Special" environment variable and Birdie

was used with VisualPrefs' "Don't optimize..." option turned on.

· Some menu items of the GUI editor could fail to correctly update the

contents of the "Bitmaps" text gadget in the "System" page. Now fixed.

Enhancements:

· VisualPrefs can now change the look of the ReAction iconify gadget.

· The "tbiclass" API now supports a new attribute allowing to ask an image

about its inner dimensions (useful for more accurate rendering of

additional imagery inside the image). See the "VP-Developer.doc" file for

more details.

· The version string of the VisualPrefs and GUI executables now reports the

specific CPU the program is optimized for (except for the 68000 version).

1.5b (25.7.99)

Small update:

· It seems that the new semaphore system, while working better for most

people, causes some problems on a few specific configurations.

So you can now choose whether to use it or not: by setting a bit in the

"Special" environment variable you tell VisualPrefs to keep handling

semaphores the old way.

1.5a (20.7.99)

Bug-fixes:

· Now VisualPrefs supports the usage of the { GTSC_Arrows, 0 } tag item

in the creation of GadTools scroller gadgets, although it's actually

an undocumented feature of GadTools and therefore shouldn't be used.

· Reworked the internal semaphore handling in order to avoid some rare

(but possible) deadlocks at boot time and/or at the re-opening of the

Workbench screen.

· The wrong pens were being used to draw the frame of selected screen depth

gadgets in "Simple" or "Filled" style. Now fixed.

· Now the GUI editor refuses to load anything which isn't an IFF preferences

file.

· Included new select.gadget which fixes a bug that caused it to ignore

the NG_HIGHLABEL flag when rendering non-centered gadget labels.

Enhancements:

· Now VisualPrefs consumes significantly less CHIP memory when using bitmaps.

· By setting a bit in the "Special" environment variable you can now prevent VisualPrefs from adjusting the results of the IM_FRAMEBOX method of "frameiclass" (usually done to better reflect the border thickness).

· By setting a bit in the "Special" environment variable you can now prevent VisualPrefs from using any CHIP memory at all, if you're using CyberGraphX and have the PLANES2FAST variable set to "1". If Picasso96 supports a similar feature, this should work with it too, although it wasn't tested.

1.5 (25.5.99)

Bug-fixes:

· Corrected an error in the GUI editor which prevented the cycle gadget used to set the look of "pop-up" highlighted items from working properly.

· VisualPrefs could fail to free some memory at the closing of the Workbench screen. Now fixed.

· Now VisualPrefs handles much better the palette of two-color screens.

· GadTools string gadgets could sometimes have incorrectly sized frames. Now fixed.

· Setting the preferences to the "Four-color" mode on the Workbench screen, then visiting another screen, doesn't cause anymore the loss of Workbench's multicolor settings due to them being overwritten with the four-color ones.

· Edges of pop-up menu drop shadows have now the correct shape when the "round edges" option is active.

· The dithering for normally disabled gadgets and for the background of GadTools proportional gadgets was being rendered with an incorrect color in the GUI editor's preview area. Now fixed.

· Fixed a bug in the screen depth gadget handling code which could cause deadlocks in certain situations.

· Corrected some details of the preview display in the GUI editor.

· If a screen using the template couldn't be opened, VisualPrefs was increasing the template usage counter anyway, and never decreasing it afterwards. Now fixed.

· Added a workaround for a conflct with NewIcons which caused window titles to be written more than one time and slightly out of phase when NewIcons' "Outline" option was selected.

· Added some stack-checking code in the "frameiclass" patch: now the custom routines of VisualPrefs aren't executed if the stack is too low. This prevents the input.device from crashing for a stack overflow when clicking on some (fortunately very rare) stack-hungry GUIs.

· Now the screen titlebar's bright edge attempts to repair itself whenever it's damaged by Directory Opus titlebar messages.

· Added a workaround for a bug in WordPerfect which caused the program's
requesters not to be displayed with VisualPrefs running.

Enhancements:

· Added support of external bitmap images for system gadgets!
You can also redefine the symbols of checkbox and cycle gadgets.
See the new section to learn how to use this feature, as well as the
"Sample.ilbm" picture and the sample bitmap set found in this archive.

· Now the button frame style and inner symbol style for system gadgets have
become independent from one another, and can thus be chosen separately.
A number of new frame styles have been added, which can be used with any
of the usual symbol styles (Standard, XEN, Future, Flat).

· Added a new pen (Inactive Glyph) for selection of the color of system
gadget symbols in inactive window borders.

· Added an option to use the Fill Shine and Fill Shadow pens for the edges
of selected frames (when "Alternate look..." is off).

· Now selected XEN frames are filled with the Foreground pen (rather than
the Background pen) if the "Don't fill selected frames" option is not set.

· Now the "Intuition is attempting to reset the Workbench screen..."
requester issued by VisualPrefs should always disappear automatically
when all the non-Workbench windows have been closed, just like the
original IPrefs requester does.

· Now the GUI editor tries to open a cloned screen even if the requested one
is already open, if the already-existing screen is too small.

· Added a new environment variable "VisualPrefs/CopyGTText": set it to "1" to
tell VisualPrefs to copy the label text of GadTools button gadgets, rather
than simply referencing the original string. This may come useful if you
use some programs which incorrectly reuse the same string pointer when they
create a list of GadTools gadgets.

· Now GadTools string gadgets can have a special look when in disabled state.

· Now VisualPrefs can (optionally) change the system standard DrawInfo pens
on all screens, rather than only on the Workbench screen. This works also
for the template.
Also, when doing so, a checkbox in the "Color locking" page allows you to
choose whether to have all system standard pens always automatically locked
(like Intuition normally would do) or to selectively lock them yourself by
hand (like you usually do with custom pens).

· The pop-up patch for cycle gadgets can now optionally draw the menu
panel directly on the screen's bitmap, rather than opening a window.
This enhances the "smoothness" of pop-up menu operations.

· Pop-up menus of cycle gadgets can now have a nice zoom-like effect when opening and closing, with a configurable duration.

· New SCREENBARFONT tool type for the GUI editor, allowing to use the current screen's font for screen and window titlebars drawn in the preview area, even if using a different font for the rest of the editor. This way you can see the titlebar gadgets exactly in the size they would have on the current screen, rather than the size they would have on a screen using the editor's font.

· When saving the template, the GUI editor now displays a proper requester with a number of gadgets allowing to choose all template options at once, rather than having to go through a series of simple EasyRequesters.

· A new "Allocate new color" option for the template, which allows for more precise color selection than "Find best match", creating the color if a suitable one doesn't already exist (it doesn't lock the pen, however).

· A new "Lock pens" option for the template, allowing to lock all the GUI pens on screens which the template is applied to (only if the template uses the saved values for pen assignment, as otherwise it isn't necessary).

· A new "Bitplane threshold" setting for the template, to determine a minimum number of bitplanes under which the pens are given simple default values regardless of what method was selected for pen assignment. This is especially useful with pen color remapping, which is usually ok with deep screens but gives poor results on screens with few colors.

· A new "Allow nameless screens" option for the template, which tells VisualPrefs to apply the template even to screens having no title.

· Reworked some internals of the GUI editor in order to make it somewhat quicker, especially when refreshing or resizing its window.

· Now the "Open..." menu item is no longer disabled on screens other than the Workbench one, allowing you to load preset preferences files on any screen. Furthermore, when loading a preset containing just one screen definition, you can now choose to apply it to the screen you're editing rather than to add it to the screen list or to replace the list with it (however, note that applying doesn't affect pen assignment).

· By great demand (about three people, one of which didn't ask me directly) VisualPrefs is now able to make the window sizing gadget thinner than the standard 18 pixels, therefore narrowing also the right window border. However, as Intuition does NOT really support such a thing, in order to achieve this the right window border needs to be quickly redrawn by the patch upon opening. Hopefully, on fast enough systems this won't be too noticeable; if it is, just don't use this feature.

· A new menu item in GUI allows you to launch your preferred palette editor.
The default is SYS:Prefs/Palette, but you can override this by using the
new PALETTE tool type to indicate a different editor (like FullPalette).

· The GUI editor now reads some settings (currently SCREENBARFONT and PALETTE)
from its icon if you don't specify them on the command line when starting it
from the Shell.

· New "Lock all" and "Free all" gadgets added to the "Color locking" page
for your convenience.

· Slightly bettered the aspect of arrow symbols in window borders with the
"Borderless" system frame style.

· A recent version of FPPrefs (40.22) included for people who use FullPalette
and haven't yet downloaded the latest release from Aminet. Versions prior
to 40.20 don't work correctly with VisualPrefs 1.5.

· Now screen titlebars can have a different height than window titlebars!
If you have square titlebar gadgets, you can force the screen depth gadget
to keep the same width as the window depth gadget by setting a bit in the
"Special" environment variable .

· By setting a bit in the "Special" environment variable you can now have a
new global look with thicker 3D edges for XEN button frames.

· Now GadTools gadgets in XEN style use the Text pen (rather than the Fill
Text pen) for their label when they are in "selected" state. This preserves
the correct XEN look regardless of what color you choose for window titles.

· Two new environment variables , "SafetyLevel" and "MinStackSize", which
allow to choose how VisualPrefs must behave in low stack conditions within
its patches to system BOOPSI classes. If needed, VisualPrefs is now able to
temporarily switch to a larger stack when executing BOOPSI classes code.
Note well: this feature must be regarded as experimental!

· Now the preview area of the GUI editor should be displayed a bit faster
on hicolor and truecolor screens.

· Added information about the new VisualPrefs mailing list.

1.4 (23.7.98)

Bug-fixes:

· Added a workaround for a bug in graphics.library/BltMaskBitMapRastPort()
which caused system gadgets to appear as garbage when Birdie compatibility
was enabled on interleaved 8-bit screens.

· Added a workaround for a bug in CyberGraphX which causes line drawing to
sometimes produce different results than expected when the PLANES2FAST
option is turned on. This was affecting especially the shape of XEN arrows.

· Corrected a couple of errors in the appearance of horizontal arrows in

the "Standard" style.

· Now window titles should no longer be drawn twice (and out of phase) when the PowerWB patch is running.

· Fixed a bug in the rendering of window titlebars when Birdie is running.

· Now the GUI editor always checks the version of catalog files.

· Fixed a bug which caused the Draw() function to be called with an illegal -1 coordinate when applying the KingCON patch.

· Now VisualPrefs checks if a screen can actually be closed before removing its patches. This should solve most problems of "GUI reverting to normal" which happened with programs such as WebDesign and Cinema4D.

· When using MagicMenu in non-blocking mode, window titlebars covered by a menu no longer have their title refreshed at the wrong position after the menu closes.

· Cycle gadgets in "Future/Flat" style had their symbol rendered with a wrong color when pressed if the gadget frame didn't use the Fill pen for its background in "selected" state. Now fixed.

Enhancements:

· Now cycle gadgets can display a pop-up menu when pressed! The behavior of the pop-up menu can be configured with the GUI editor (sorry, no preview yet). You can even set a delay for the pop-up menu to wait before appearing, so that with "quick" clicks the cycle gadget will still behave like usual. Please read the guide for more information.

· Disabled GadTools button and cycle gadgets can now be displayed with a new 3D look (without ghosting and with their text recessed). You can also choose the colors to be used for this.

· GadTools button and cycle gadgets can now "shift" their contents when pressed (like scroller arrows already did).

· Now "Future" style border gadgets can optionally be made to only react to clicks made within the actual "button" image, rather than the ones made on the whole gadget hit box.

· You can now choose a XEN look for GadTools gadgets without affecting all other 3D frames in the system (although of course you can still do it the old way, too).

· Now you can have compatibility with Birdie on all screens, not just the Workbench one. The way to configure it HAS CHANGED, please read the guide! It now also involves setting an option in the GUI editor.

· Now the titlebar of screen can be made fully 3D, by adding the missing "bright" edge at the left and top sides. A new pen (Bar Contour) has been added to this purpose.

· You can now set the color of inactive window titles.

· The IM_FRAMEBOX method of "frameiclass" now reflects the real thickness

of the frame borders.

· The "PatchWB" program by Thomas Richter has been added to the VisualPrefs

distribution archive; it should fix the annoying Workbench screen re-open

lock-up problem once and for all, by killing its real cause. Thanks Thomas!

· At last, you can install a proper 68060 version of VisualPrefs and GUI.

· Now the installation script allows you to choose whether to install icons

or not, so you can avoid overwriting your existing icons.

· The GUI editor now embeds a version number in the preferences file when

saving. This will make much easier in future releases the addition of new

features without loss of compatibility.

1.3c (30.12.97)

Small update:

· Fixed a bug which caused Enforcer hits at the opening of Workbench windows

when the PowerWB patch was running.

· DirOpus iconify & padlock gadgets should now be patched whichever window

they appear in; previously some windows were missed by VisualPrefs.

· Now the GUI editor saves a "palette.prefs" file with default settings if

one doesn't exist.

1.3b (15.12.97)

Small update:

· Compatibility with Birdie in hi-color and true-color modes was broken

due to the previous bug-fix. Now it works again.

1.3a (12.12.97)

Small update:

· Fixed an odd bug which prevented VisualPrefs from running on most V39

(Kickstart 3.0) systems.

· Slightly modified the documentation.

1.3 (26.11.97)

Bug-fixes:

· Fixed a bug that caused the YAM mailer to display fake error messages.

· Now the iconify gadget of the DirOpus text reader window is also patched.

· The DirOpus screen title wasn't always vertically centered. Now fixed.

· Removed an Enforcer hit with DirOpus 4.

· Now the additional titlebar gadgets (such as the MUI ones) have the correct

colors on Picasso96 hicolor and truecolor screens.

· The previous fix for the YAM problem, found on my home page, did cause

address errors on plain 68000 CPUs. Now fixed.

· With big titlebar gadgets, sometimes the window title covered part of the gadget at its immediate right. Now fixed.

· Saving the preferences doesn't corrupt anymore the titlebar of some screens if they have an increased titlebar height.

· Now deadlocks at the reopening of the Workbench screen when saving the preferences should be less likely to happen.

Enhancements:

· Now VisualPrefs can be made nearly 100% compatible with Birdie! See the documentation to learn how to obtain this.

· Now the KingCON patch also works for RAW: and KRAW:.

· Now VisualPrefs also patches the Scalos iconify gadget image.

· Added some developer documentation about the "tbiclass" image class.

1.2a (22.10.97)

Small update:

· Fixed a bug in the installation script.

· The "Flat" screen depth gadget image now uses the Bar Detail pen if the Glyph pen and the Bar Block pen have the same value.

· Fixed a bug in the look of the inactive "padlock" image in the "Standard" style.

1.2 (18.10.97)

Bug-fixes:

· The patch for the ClassAct iconify gadget didn't work with the latest release of ClassAct. Now this is fixed.

· The workaround for the V39 gadtools.library bug didn't work very well, so I found another one.

Enhancements:

· Now VisualPrefs patches DirOpus 5.5+ titlebar gadgets. Only tried it with Magellan, but it should work also with previous releases. Hope you like the new "padlock" image...

· Renamed "iconifyimageclass" to "tbiclass" to comply with the new API of ViNCEd. Now VisualPrefs should be 100% compatible with ViNCEd (3.53+).

· Now you can choose to install a 68020/030 or 68040/060 version of VisualPrefs and GUI.

· Some optimizations to the code of certain patches.

· Of course, this release also contains all the changes introduced in release 1.1, which wasn't released on Aminet (see below).

1.1 (3.10.97)

Some bug-fixes:

· No more crashes with MUI "screen close" gadgets.

· The screen titlebar doesn't get corrupted anymore under CyberGraphX 3
with the SUPERLAYERS option on.

· The GUI editor now correctly fills the bottom scroller's hole in the
preview area if you select the "Fill recessed areas..." option.

· Now the GUI editor allows for fonts with an height smaller than 7 pixels;
the new limit is 5 pixels.

· GadTools arrow gadgets didn't get patched if their width or height was
smaller than 10 pixels; now the limit is 9 pixels.

· The image of CheckBox gadgets sometimes did overwrite part of the border.

· Added a workaround for a bug in the V39 gadtools.library concerning the
displaying of the contents of number gadgets. Hope it works!

· The "Global" menu is now correctly updated when you select "Last saved"
from the "Edit" menu.

Enhancements:

· The rendering of MUI titlebar gadgets should now be somewhat faster.

· Added a patch for MagicMenu 2.19+ allowing to vertically center the
menu titles in the screen bar.

· Added the possibility to inhibit the use of the Template.

· Added public "iconifyimageclass" image class for use by programs such
as ViNCEd.

1.0a (26.7.97)

Small update:

· Fixed some bugs in the installation script.

· Added a few more safety controls in the GUI editor.

· There were some cases in which the Workbench screen would lock up
if closed and reopened. Now this shouldn't happen anymore.

· Slightly modified the documentation.

1.0 (23.7.97)

First Aminet release.

Beta releases #1 to #7

Private beta-tester releases.

1.29 The VisualPrefs users mailing list

THE VISUALPREFS USERS MAILING LIST

=================================

A mailing list is available for the users of VisualPrefs. You can subscribe
to this list in order to talk with other users of VisualPrefs, to share
ideas about GUI configuration or ask for help about small common problems.

This can be especially important in periods like summertime, when I'm

unable to reply to questions sent to me by e-mail.

To subscribe to the list, just send an e-mail to

vpml-subscribe@onelist.com

without specifying anything particular in the subject or body text.

You might be asked to confirm your request with a second e-mail, depending on

whether or not you are already a registered user of ONElist.

You can also subscribe to the list by following the link you find on my

web page, in the "Projects" section.

To unsubscribe, you can send an e-mail to

vpml-unsubscribe@onelist.com

or just ask me to remove you from the list.

Have fun with the VisualPrefs mailing list!

1.30   Some information about the author of VisualPrefs

AUTHOR INFORMATION

==================

Massimo Tantignone

Via Campagnoli, 4

28100 Novara (NO)

ITALY

E-mail: tanti@intercom.it

WWW: http://www.intercom.it/~amigaws