NotesAndHints

COLLABORATORS

	TITLE : NotesAndHints		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		September 19, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

NotesAndHints

1.1 NotesAndHints

```
              Notes and Hints : Questions, which might be "asked frequently"
 (1.7.94)         in the future ...




                    SuperView-Library sounds good, but how shall I use it ?

                    How much Memory does this program eat ?!

                    Problems with converting HAM Files

                    Problems with displaying 24 Bit Files

                    Problems with converting 24 Bit Files

                    Programmers note for releasing own SVObjects / SVDrivers

                    Where - the hell - is this JPEG.svobject ?

                    One of the advertised features does not work ...

                    Philosophy ... :-)

                    Copyrights
```

1.2 usagenote

```
 SuperView-Library sounds good, but how shall I use it ?
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   Well, I guess you belong to this group of people, who downloaded
   SuperView-Library from a BBS, but did not download an appropriate
   Viewer-Program also ?
```

OK, this is not as worse as it sounds : The people, who instead
downloaded the Viewer-Program without the Library will perhaps
be in a much more worse situation ...

Back to the problem :
SuperView-Library is FREEWARE and the most appropriate Viewer-Program
for it currently is SuperView (SHAREWARE).
As with Version 3.x it has as many features as a Viewer- and Conversion
Program for SuperView-Library can have.
You should get it as soon as possible !

But for a simple way of testing the library it should also suffice,
if you use one of the supplied Example-Programs (Directory "Programmers").
There are e.g. "SimpleView" for Workbench users and "MicroView" for
CLI/Shell users.

Last not least all of the programming stuff is included, so that you
might get inspired to write your own Viewer-Program ?!

1.3 memprobs

How much Memory does this program eat ?!
~~~~~~
  Well, er, this is a good question, because there are no fixed
  limits ...
  superview.library and its attached SVObjects and SVDrivers
  request as much memory as they need and there's no virtual memory
  handling system or such a thing.

  But nearly no CHIP memory is needed for most of the actions, so
  1-2 MB CHIP mem should suffice for almost all pictures (if they
  are not sized about 2048x2048x24, I mean, what do we talk about ?!).

  Most of the actions are performed in FAST Ram if possible, so you
  should have at least 2 MB Ram at all (1 CHIP, 1 FAST or 2 CHIP),
  better 4 MB (2 CHIP, 2 FAST) or more.

  An example : When trying to display a 1280x1024x6 picture on my
  A4000/030 with 4 MB, I have to free the last memory resources to
  get the picture displayed. Otherwise I will get it loaded, but not
  displayed ...

  The large memory usage results out of the buffering-technique, which
  is used in the superview-Library system.
  This technique increases memory usage by using large (full-sized)
  buffers, which often do exist twice e.g. for conversion from
  chunky-pixel to bitplane format or vice versa.
  But this increases SPEED !!
  There are perhaps no "real" multimedia-systems out there, which have
  less than 6 - 16 MB and why shouldn't we use this memory ?

  On a 68040/68060 system you should see superview.library flying ...

  OK, back to earth ! Here's what you can do to make more memory available
  to superview.library :

- first of all, check if there are some background-programs running,
  which you do not really need (especially Commodities).
  Remove them !
- call "avail flush" in the Shell or select "flushlibs" in the
  WB-debug menu (available when started with "LoadWB -debug")
- decrease the "AddBuffers" values of drives, which you do not
  really need (e.g. DF2, DF3, PC0, ...)
- last not least : close Workbench, stay in the Shell
- start your favorite Viewer or Conversion program (e.g. SuperView)
- again : call "avail flush"
  (this time also all unused SVObjects and SVDrivers will be flushed
   out of the memory, so that we later only will have the needed ones
   in Ram)

If you tried all of the above, and memory still does not suffice,
you should buy some more SIMMs at your local Computer Shop ...

1.4 hamprobs

```
Problems with converting HAM Files
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 The Problem
 -----------
  You have problems converting HAM6 or HAM8 files into GIF, PCX or
  such "alien" File Formats ?

 The Reason
 ----------
  Well, this is because those FileFormats do not preserve any ScreenMode
  information, so that a HAM6 file becomes just a plain 64 Color BitMap
  graphic and HAM8 turns to a 256 Color graphic.
  (And this will look really UGLY without doing any further dithering ...)

 The solution
 ------------
  There are two possible ways :
 a) only convert to FileFormats, which preserve the ScreenMode information
     (currently there are only IFF-ILBM and IFF-ACBM).
 b) use a special conversion program for converting HAM-graphics into
     plain IFF-graphics (dither them down to 32/256 Colors) and then
     convert them with SuperView.

 Future Solutions
 ----------------
  It is not planned, to integrate dithering-routines into SVObjects or
  SVDrivers for converting HAM graphics into more common formats.
  Nevertheless this might be introduced with the (magic and mysterious)
  SVOperators.
```

1.5 24bitprobs2

```
Problems with displaying 24 Bit Files :
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 The Problem
 -----------
  "All of my attempts in displaying my really nice 24 Bit graphics with
   AGA.svobject resulted in really ugly colors. What shall I do ?"

 The Reason
 ----------
  AGA.svobject does not interpret or compute the best possible gamma
  correction table (256-color map for "dithering" the picture down),
  but it uses the default ColorMap of AGA-Screens instead.
  This produces quite good results ususally, but may sometimes not
  be satisfying, especially for real "graphic freaks", which might
  expect better solutions and results.

 The solution
 ------------
  There's one way to pass this by :
  When reading JPEG-graphics, you may select (via a ControlPad preferences
  file), whether the output should be done in 8 or 24 Bit.
  If you select 8 Bit, the JPEG.svobject will do a very good down-sizing
  on the 24 Bit graphics : The results are much better than from AGA.svobject.
  The resulting graphics will then be displayed just like a common 256 color
  graphic via AGA.svobject.
  Here's my solution, which is just for people, which will often display
  24 Bit graphics on AGA, but do not plan to edit them anymore (nevertheless
  only do the following after you HAVE MADE BACKUPS of all files !) :

   1) Load the 24 Bit File into SuperView
   2) Select "JPEG" (JFIF) as output file format
   3) Save the graphics
      (requires at least V2.3 of JPEG.svobject)
   4) Select "8 Bit" for JPEG-output (is default)
   5) Load the new JPEG-File
   6) Select an other output file format, e.g. ILBM
   7) Save the graphics

  Now you have a very well dithered copy of your 24 Bit graphics.
  But note, that JPEG-compression is "lossy", so that you may not
  re-construct the original picture from it, even if you'll select 24
  instead of 8 Bit output (so remember to make backups).
  Nevertheless the resulting JPEG-image usually does not show any
  visible differences compared against the original file.
```

1.6 24bitprobs

```
Problems with converting 24 Bit Files :
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 The Problem
 -----------
  You have problems converting 24 Bit files into other 24 Bit files
  (e.g. from 24 Bit-Targa to 24 Bit-PCX) ?
```

```
The Reason
----------
 Not all SVObjects, which support reading of 24 Bit Files do also support
 writing of 24 Bit Files.
 SVObjects, which support writing of 24 Bit data yet are :

   - JPEG (since V2.3)
   - Targa
   - BMP
   - SunRaster
   - SVO

 The reason is, that I just introduced the saving of GfxBuffers
 for all SVObjects (since V6.1), so that this is the first time, that
 the SVObjects might be confronted with the saving of 24 Bit data
 (before this point there were only Screens with 8 Bit ColorDepth a
 possible source for writing).

 The solution
 ------------
 This feature may be available for more SVObjects soon (don't ask me to
 explain, what "soon" means in this case ...)

 A temporary solution might be to use other 24 Bit programs for
 conversion, if you don't want to use one of the file formats, which
 are already supported.
```

1.7   prgnote1

```
Programmers note for releasing own SVObjects / SVDrivers :
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 The Idea
 --------
  When writing your own SVObjects and SVDrivers, and distributing
  them via Nets or MailBox/BBS, please name your archives in the follwing
  way :

           "svo*..*xy.EXT"    for SVObjects
    and    "svd*..*xy.EXT"    for SVDrivers

    where - "*..*" stands for the library's name          (e.g. "dummy")
          - xy stands for the version and revision number (e.g. "11")
          - EXT stands for the archiver's extension        (e.g. "LHA")

 The Reason
 ----------
 The space for describing files on BBS is not very large (let's say
 about 20 characters) and most people do not really read the attached
 mails or readme-files usually.
 So it might be useful to see right from the FileName :
 "Yes, it's a SVObject" (svo#?) or "Yes, it's a SVDriver" (svd#?).

 An Example
 ----------
 Version V2.1 of my JPEG.svobject is distributed as :
```

```
"svoJPEG21.LHA"
```

1.8 jpegnote

Where – the hell – is the JPEG.svobject ?
~~~~~~
  The JPEG.svobject is distributed in a separate archive, due to the
  fact that most people don't like to download megabyte-sized archives,
  just to find out, that they contain hundreds of kilobytes they do
  not need actually.

  Those who need JPEG-Support should look out for an archive called
  "svoJPEGxy.LHA" where "x" stands for the version and "y" for the
  revision number.

  JPEG.svobject requires at least an 68020+ and allows 8 Bit and 24
  Bit output. If you only have ECS, you will perhaps not get any results
  on the Screen. You will at least have to use the AGA-SVDriver (for
  well-dithered 8 Bit output) or EGS-SVDriver (for 24 Bit output or
  dithered OCS/ECS/AGA output).

1.9 bugnote

One of the advertised features does not work ...
~~~~~~
Well, sometimes it happens, that the elimination of one minor bug
causes two major bugs, so that a version of a SVObject or SVDriver
is actually much less useful, than previous versions, although a bug
has been fixed.

If you sometimes face such a problem, just install the old version of
this specific SVObject or SVDriver and keep the rest of the new
installation intact. SVDrivers and SVObjects are fully compatible
from the first version upto the current version.
Of course there might be exceptions, but those would be again Bugs ...

If you do not have any older versions to replace an existing SVObject
or SVDriver, you have to decide, whether you'll remove it completely
and just wait, until the next version comes out, or if you want to
use the buggy one nevertheless.

Besides : if you'll decide to report the bugs to me, they'll perhaps
         be eliminated sooner ...

But what I'm talking about ?
I guess, this version is totally free of bugs ... ;-)

1.10 philosphy

```
"Bei Risiken und Nebenwirkungen essen Sie die Packungsbeilage
 und @$_!@*# Sie Ihren Arzt oder Apotheker"

(I do not feel able to translate this ^ into any other languages ...
 It's just a parody on a slogan, which has to be placed at the end
 of german commercials, if they offer pharmaceutical products, which
 might cause physical problems for the consumer ...)


Murphy's laws of working with BitMapped Graphics :

 (excerpt)

 No. 23 : When you've integrated the fastest, smartest C routines
          for Huffmann-decoding, which you were able to find in the
          whole public domain, you'll see, that there's someone out
          there, which wrote a 20 times faster version in 68030 assembler.

 No. 55 : After 20 nights and days of un-interrupted programming
          it'll happen, that in the 21th night a mysterios bug
          appears, which will make all of your previous work obsolete.

 No. 64 : If you're making great usage of the OS and you've integrated
          all features, which this version offers to the programmers,
          it will take about half an hour and all BBS will post the
          message, that the new OS 999.1 is out and that it contains
          support for almost any BitPlane graphic formats in the
          whole wide world.

 No. 99 : If you've written a program, that writes BitMaps, you will
          perhaps have forgotten that sometimes the data has to
          be padded to giga-word boundaries, to make reading of
          the data faster on the new H.A.L. 5000.
```

1.11 copyrights

```
    Copyrights
```

Some of the mentioned names or products above may be copyrighted by
companies or trademarks of companies.