

NotesAndHints

COLLABORATORS

	<i>TITLE :</i> NotesAndHints		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	NotesAndHints	1
1.1	NotesAndHints	1
1.2	usagenote	1
1.3	memprobs	2
1.4	datatypes	3
1.5	hamprobs	3
1.6	24bitprobs2	4
1.7	24bitprobs	5
1.8	prgnote1	6
1.9	jpegnote	6
1.10	bugnote	7
1.11	philosphy	7
1.12	copyrights	8

Chapter 1

NotesAndHints

1.1 NotesAndHints

"Notes and Hints" FAQ - V11.3 (2.4.95)

SuperView-Library sounds impressive, but...

How much Memory does this program eat ?!

What about Datatypes ?

Problems with converting HAM Files

Problems with displaying 24 Bit Files

Problems with converting 24 Bit Files

Programmers note for releasing SObjects/Drivers/Operators

Why - the hell - does this JPEG.svobject not work ?

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 4.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 SVOjects and SVDdrivers request as much memory as they need and there's no virtual memory handling system or such a thing (I introduced it with V9.9, but upto now there aren't any modules making use of it).

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 with only 4 MB, I always had to free the last memory resources to get the picture displayed. Otherwise I got it loaded, but not displayed ... (that's why I bought some more RAM ... :-)

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"). As with V4.2 SuperView also allows flushing Libraries via Menu calls.
- 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 SVOjects and SVDdrivers 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 datatypes

What about Datatypes ?

~~~~~

Well, guess what: since superview.library deals with graphics files, there might be no need to support other kinds of datatypes. On the other hand, it would be nice, to have one viewer/player for all the Datatypes and - maybe - for replacing the standard viewer.

The current implementation of Datatypes-Support inside superview-library allows it to read and/or display ANY Picture-Datatype and to link it - keeping all its functionality - into superview.library's control structures.

Putting the single keyword "ANYDATATYPES" into "ENV:SuperView-Library/LIBRARY.controlpad" enables support for datatypes other than Graphics.

Superview.library (i.e. SuperView) will then be able to display ANIMs, CDXL, WAV, and so on, if the according datatype is already in memory (that's the most important part ;-)

This enables you to replace the standard DT-Viewer of the WB in most of its facilities.

## 1.5 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 would become just a plain 64 Color BitMap graphic and HAM8 turns into a 256 Color graphic.
(And this would 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 SVOjects or SVDDrivers for converting HAM graphics into more common formats.

Any SVOjects, which do not preserve ScreenMode information will perhaps return an error when trying to save HAM graphics.

You should use "AnyTo24Bit.svoperator" to convert HAM into 24 Bit an then try saving it as a 24 Bit graphics format.

1.6 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) by default, but it uses the generic 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

-----

You may a) dither the 24 Bit graphics before displaying,  
or b) save it as a 24 Bit JPEG and then select 256 Color output.

- a) Load the 24 Bit RGB graphics into SuperView, then select "24BitToHAM" or "Dither24Bit" as SVOperator and perform an "Action". Now you may select "View" or "Save" to handle the graphics.

When using AGA.svdriver you might also add an appropriate entry to AGA.controlpad, so that any dithering operations will be done before displaying the graphics (e.g. "24BITOPERATOR=24BitToHAM" or "24BITOPERATOR=Dither24Bit").

- b) 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 usually well-dither the 24 Bit RGB data: The results are much better than from AGA.svobject. The resulting graphics will then be in 256 Colors. Here's a quick-and-easy description how to do this (don't forget to make backups before):

- 1) Load the 24 Bit File into SuperView
- 2) Select "JPEG" (JFIF) as output file format
- 3) Save the graphics
- 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.

## 1.7 24bitprobs

Problems with converting 24 Bit Files :

~~~~~

The Problem

You have problems converting from one 24 Bit file format into other 24 Bit file formats ?

The Reason

Not necessarily all SVOjects, which support reading of 24 Bit Files do also support writing of 24 Bit Files.

SVOjects, which support writing 24 Bit data yet are :

ILBM	JPEG	Targa
BMP	SunRaster	YUVN
PCX	FBM	PNM
SVO		

The reason is, that a) 24 Bit support has been added at last and that b) not all file formats do support easy (or even any) support 24 Bit image data.

The solution

This feature may be available for more SVOjects soon (if possible

and useful)...

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.

1.8 prgnote1

Programmers note for releasing own SVOBJECTS/DRIVERS/OPERATORS:

~~~~~

The Idea

-----

When writing your own SVOBJECTS/DRIVERS or OPERATORS, and distributing them via Nets or MailBox/BBS, please name your archives in the following way :

```

        "svo*..*xy.EXT"   for SVOBJECTS
or      "svd*..*xy.EXT"   for SVDIVERS
or      "svp*..*xy.EXT"   for SVOOPERATORS

```

where - "\*" stands for the library's name (e.g. "JPEG")  
 - xy stands for the version and revision number (e.g. "22")  
 - 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 SVDIVER" (svd#?)  
 and so on.

An Example

-----

For example, version 2.2 of my JPEG.svobject had been distributed as :

```
"svoJPEG22.LHA"
```

(now it is integral part of superview.library's "Bonus" directory, but there may still occur separate archives containing single modules from time to time)

## 1.9 jpegnote

Why - the hell - does this JPEG.svobject not work ?

~~~~~

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-SVDIVER (for well-dithered 8 Bit output), the EGS-SVDIVER (for 24 Bit output or dithered OCS/ECS/AGA output) or Steve's & Paul's OPAL-SVDIVER

(for 8/24 Bit Display on an OpalVision Graphics Card).

Also, you should ensure, that a directory called "VMEM:" actually does exist (see documentation).

P.S.: If you need to run 68020 programs on your well-tuned 68000 system and you don't mind some loss in speed, you may try to emulate 68020 functions by using Kamran Karimi's "Run68017" package, which can be found on AmigaLibDisk (Fish) #798. I don't know, whether this will work, but you might decide to try this.

1.10 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.

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

Example: In V9.1-9.4 of superview.library there was a slightly buggy AGA.svdriver (autoscroll deactivated), which resulted out of an other bug-fix.

There would have been three solutions: a) using ECS.svdriver  
b) using old AGA.svdriver  
(with old bug)  
c) waiting for V9.5 ...

## 1.11 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:  
(Andy's release ;-)

(excerpt)

- No. 23 : When you've integrated the fastest, smartest C routines for Huffman-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. 34 : If you try to send files via UUCP, they'll be bigger than 20 K with a likelihood greater than 99.999%.  
But only, if you don't see an other way to send them.
- No. 41 : If you make backups you'll overwrite them with the next buggy version, nevertheless.  
Guess ?
- 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. 58 : Never believe, that the mysterious behaviour of your program might not necessarily be a "real bug".  
It's ALWAYS a "really big bug".  
Even bigger than you ever thought one single bug might be ...
- 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. 77 : If you've converted all your pictures to an other FileFormat - of course without making copies, 'cause you'd like to save a lot of diskpace - you'll detect that the Writer program of the FileFormat has been slightly defective, so that the supplied Reader program is the only one, which will read your pictures then, any longer.
- No. 87 : Ever tried to port some code from XX-DOS ?  
Still working on it ?  
Never mind, it's the best way to waste your time - really !
- 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.

## 1.12 copyrights

---

### Copyrights

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

---