

SuperView-Library

COLLABORATORS

	<i>TITLE :</i> SuperView-Library	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		January 18, 2023
<i>SIGNATURE</i>		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 SuperView-Library	1
1.1 SuperView Library Documentation	1
1.2 Copyright	2
1.3 Disclaimer	3
1.4 Distribution	3
1.5 Abilities, Purposes and much more	6
1.6 Installation and Configuration	6
1.7 Overview of currently available SVOjects	9
1.8 Overview of currently available SVDivers	11
1.9 Overview of currently available SVOperators	12
1.10 Software supporting superview.library	14
1.11 superviewnote	15
1.12 sqopalnote	16
1.13 superloadernote	16
1.14 imageengineernote	16
1.15 picmanagernote	17
1.16 drafunote	17
1.17 Thanks, greetings, credits and the rest	18
1.18 How to contact the author	21
1.19 The future of superview.library...	22
1.20 Known bugs, crashes and their possible workarounds	22
1.21 Harddisk MaxTransfer Problem	23
1.22 Startup Crashes - Processor Dependent	23
1.23 Problems with specific applications	24
1.24 History	26
1.25 AmiFIG.svobject	34
1.26 ILBM.svobject	36
1.27 ACBM.svobject	38
1.28 Datatypes support	39
1.29 PCX.svobject	39

1.30 SVG.svobject	41
1.31 GIF.svobject	43
1.32 BMP.svobject	44
1.33 WinIcon.svobject	45
1.34 FBM.svobject	46
1.35 PNM.svobject	47
1.36 PNG.svobject	48
1.37 C64.svobject	49
1.38 CDR.svobject	50
1.39 IMG.svobject	50
1.40 TIFF.svobject	51
1.41 EPS.svobject	53
1.42 GhostScript.svobject	55
1.43 Targa.svobject	57
1.44 WPG.svobject	58
1.45 SunRaster.svobject	59
1.46 SGI.svobject	60
1.47 PICT.svobject	61
1.48 Pictor.svobject	63
1.49 MAC.svobject	63
1.50 JPEG.svobject	64
1.51 PCD.svobject	68
1.52 FastILBM24.svobject	69
1.53 YUVN.svobject	70
1.54 DEEP.svobject	71
1.55 QRT.svobject	72
1.56 C-Source.svobject	73
1.57 UtahRLE.svobject	74
1.58 CyberDataType.svobject	75
1.59 ECS.svdriver	75
1.60 AGA.svdriver	77
1.61 CyberGraphics.svdriver	78
1.62 EGS7.svdriver	80
1.63 PicassoII.svdriver	82
1.64 OPAL.svdriver	84
1.65 Retina.svdriver	86
1.66 PICASSO.svdriver	88
1.67 MERLIN.svdriver	89
1.68 XOR.svoperator	90

1.69	24BitToHAM.svoperator	91
1.70	Crop.svoperator	92
1.71	Dither24Bit.svoperator	92
1.72	HilbertDither256.svoperator	93
1.73	AnyTo24Bit.svoperator	94
1.74	ExtractGrayScales	95
1.75	ExtractRed	96
1.76	ExtractGreen	96
1.77	ExtractBlue	97
1.78	TopToBottom	98
1.79	LeftToRight	98
1.80	Rotate	99
1.81	RotateFree	99
1.82	Scale50	101
1.83	CallPNM	102
1.84	OptimizePalette	103
1.85	PaletteDither.svoperator	104
1.86	Literature used during development of this program	105
1.87	Credits	105
1.88	Requirements for the SuperView.library Package	111
1.89	NotesAndHints	112
1.90	usagenote	113
1.91	memprobs	113
1.92	hamprobs	114
1.93	24bitprobs2	115
1.94	24bitprobs	115
1.95	prgnote1	116

Chapter 1

SuperView-Library

1.1 SuperView Library Documentation

superview.library V13.2

- Freeware (Licenseware) -
product-specific Licenseware

Any usage from and by other programs without an
explicit license is strictly forbidden (see "Distribution").
Ask for licenses.

© 1993-96 by Andreas R. Kleinert. All rights reserved.

A PERSISTANT SOFTWARE (PS) PRODUCT.
(persistent - resistant -> persistant)

This program has been written under OS V3.1 and is therefore
fully compatible. It needs OS V2.04+.

Release Date: 18.6.1996

Copyrights and legal stuff

Credits and even more legal stuff

Disclaimer

Distribution

How to contact the author

Requirements

Installation

Short: Purpose and Abilities

Supporting Software

SVObject Descriptions/History
SVDriver Descriptions/History
SVOperator Descriptions/History
Known bugs and workarounds
Notes and Hints
Possible future enhancements ...
History
Thanks and Greetings
Used literature for developing
 External Links in this Directory

About Power-Brei (German Disk Magazine)
 More Projects running
 Online Support Sites
 SuperView in the Press
 SVPrefs documentation
 ControlPad enlistment
 German ReadMe File

 _ //
Only \X/ Amiga makes it possible!

SuperView WWW page: <http://members.aol.com/tfischb/main.htm/>

Die CHAOS-Theorie:

"Dabei geht es zum Beispiel darum, den verdammten Schmetterling zu finden, dessen Flügelschläge die vielen Stürme in letzter Zeit verursacht haben."
(Terry Pratchett in "Total verhext")

Dedicated to all people, who actually READ docfiles ;-)

1.2 Copyright

The superview.library and the distributed files - e.g. the documentation files - are (C)opyright 1993-96 by Andreas R. Kleinert.
All rights reserved.

(For some files there may additional or substitutive Copyrights take place, which then are stated locally within the documentation or via local reference to "Credits".)

The following usage and license conditions are announced for all parts of the distribution, which means SVObjects, SVDivers, SVOperators and all other files.

The usage and distribution of superview.library takes place under the concept of product-specific Freeware (Licenseware).

For more details on distribution rules and developers restrictions, you have to see paragraph "Distribution".

Please note:

- * Should any of the listed terms and clauses within this document not be valid in conjunction with the law of certain countries this does not affect the validity of the other clauses.
- * Some of the mentioned names or products within this or other documents may be copyrighted by companies or trademarks of companies or persons.
- * This software is based in part on the work of the Independent JPEG Group (concerning JPEG.svobject).
- * Some parts of this software do make use of the LZW (de-)compression algorithm, which is protected by patent claims in some countries (in Germany, algorithms as such are not protectable, especially when these have been published for the public).
When using licensed versions of this library with commercial programs or on a for profit-basis, you may also have to license this algorithm (take a look at web page <http://www.unisys.com/>).
The author of superview.library does not feel responsible for offenses of third party people against that (concerning GIF.svobject and parts of TIFF.svobject).
If there are any legal doubts - YOU, as the user, must use the supplied and explained methods to disable GIF.svobject and parts of TIFF.svobject to be allowed using the library...

1.3 Disclaimer

The author takes no responsibility for any results of the use of this program.

This software is provided "AS IS" and there is no warranty of any kind, so that you use this software at your own risk.

1.4 Distribution

As with V11.8 of the Library the following distribution conditions take place:

Distribution in Common

~~~~~

The Library must not be distributed isolated, detached from any specific program package, whose author has licensed the library.

If the licensed library is included within the licensing program's program distribution, the distribution conditions of this program take place as long as these do not explicitly contradict to the disposals described in the following text.

Licensed programs are not restricted to have any special legal copying conditions, so the Library may appear together with any form of Software, like e.g. Freeware, Shareware or commercial programs, perhaps with exception of "pure" Public Domain.

## Restrictions

~~~~~

The program SuperView-Library in this version is product-specific Freeware (Licenseware), which at first means, that a single distribution of the Library is as well forbidden as an unauthorized distribution together with programs of non-licensees.

Also, as with V11.8, the Library MUST not:

- be included into other program's distributions without my explicitly written permission
- be copied as a single package without being directly related to a specific program
- without a license be copied and also not be used directly or - via any tricks - indirectly

Licensing

~~~~~

There is no way of automatic licensing via "agree-to-the-above-terms". Any license since V11.8 has to be given in written, non-electronic form. Interim-licenses may be promised via phone, email, etc but have to be confirmed in paper form.

The only one, who is authorized to write and send licenses or promise licensing is the author of superview.library, which is Andreas R. Kleinert.

If you want to license the library, please send me the following information:

STATUS - what's the status of your planned or actually existing program ?

Main categories are Freeware, Shareware, Commercial.

WHO - what's your (or your companies') name, address and phone number, maybe also email adress ?

WHAT - what kind is your program of ?  
Not any programmer of any kind of program will get a licence.

I am not going to support programs, which may concern my vital interests by being direct competitors of other programs where I am involved as a participator, depending on the success of the programs. [ ;-) ]  
On the other hand there are various ways to synchronize programming efforts and to come to a solution, which is acceptable for both sides.

PLANS - according to the type of your program you might also tell me something about your plans for future versions of the program as long as it does concern a fundamental change in the program's concept and function (competitor)

SAMPLE - would be nice to see a copy of the previous, beta or final version of the program

PAYMENT - are you gonna pay for the licence ?

Freeware authors will not have to pay anything, if they get a license. Shareware authors might. Commercial programs will perhaps.  
Tell me what you think, then we'll talk about it.  
Fees aren't expensive in my opinion.

After Licensing

~~~~~

Each license is specific to the licensee and his wishes.

Common to all licensees is the following:

- The Right to use the unmodified library within the program or program-group it has been licensed for
- Access to not publicly documented functions
- The Right to distribute the library together with your program
- The Right to integrate the library into the concept of your program as long as no basic parts have to be changed

- The Obligation to respect the distribution conditions and to add a ReadMe file and documentation remark reflecting this to the program
- The Obligation to send me a copy of the final, registered or register-able version of the program using superview.library

If there are more questions, please don't regret to ask me, because just asking doesn't cost you anything.

Developer Note:

Developer sources are available for C, Oberon2 and PCQ-Pascal, where E is planned and Asm is easy to derivate from the others (FD-files are available).

Not in all cases the newest versions are available (V9-V12; C is always up-to-date).

1.5 Abilities, Purposes and much more

The "superview.library" consists of many functions and sub-libraries, which allow quick and easy displaying, saving, converting and processing of various picture formats (IFF-ILBM, GIF, JPEG, PCX, Targa, TIFF, ...).

It has been designed for the purpose to display and process any common type of bitmap graphics as fast and as comfortable as possible on almost any Amiga hardware configuration.

External Viewer-Libraries (SVObjects), Graphic Card Drivers (SVDivers) and various Operators (SVOperators) allow easy but flexible configuration, usage and expansion.

Programmers may send me their own SVObjects, SVDivers or SVOperators, if they want to have them added to the main distribution archive or put onto the DeveloperDisk, which is available for the library.

1.6 Installation and Configuration

=====
Installation Options: Overview

- A. Automatized Installer-based Installation
 - 1. Installer procedure
 - B. Customized Installation
 - 1. Main installation procedure
 - 2. Setting Up Standard Configuration
 - 3. Plugging in 680x0 modules
 - 4. Plugging in PackerSupport
 - 5. Plugging in SQ-Operators
 - 6. Using VMEM: (Needed)
 - 7. Plugging in Ghostscript/Postscript (TM)
 - C. To Be Checked Also
 - 1. Datatypes-Support
 - 2. Other Libraries
 - 3. Additional settings
 - 4. Preferences
 - 5. LZW License Check
- =====

A. Automatized Installer-based Installation
=====

- 1. Installer procedure

Depending on the selected skill mode (novice, intermediate, expert) you will at least be asked for some of the following:

- your basic graphics engine (ECS, AGA, various GfxCards)
 - a place to install the libraries to instead of LIBS:
-

- about LZW's legal status in your country
- a place to install the catalogs to instead of SYS:locale/catalogs
- a path to assign VMEM: to (s:user-startup will be modified)
- whether SVPrefs should be copied to SYS:Prefs
- whether to install the PackerSupport stuff
- whether to immediately also install the SQ-Operators

B. Customized Installation

=====

1. Main installation procedure

Copy "superview.library" and "superviewsupport.library" to your LIBS: directory, the SVObjects to "LIBS:svobjects/", the SVDDrivers to "LIBS:svdrivers/" and the SVOperators to "LIBS:svoperators/"!

Be aware, that library names are case-sensitive and wrongly spelled libraries won't be recognized, though.

Installation of the libraries can be done to any directory, so that these may be installed separate from the system libraries. Adding the superview-Library to the system would then be managed via

```
Assign LIBS: [PATH] ADD
```

somewhere in the User-Startup.

2. Setting Up Standard Configuration

Docs/Sample_Configs contains basic configuration settings for ENVARC: and ENV: which have to be copied there to allow the library system running without problems. env_AGA contains plain AGA configuration and so on - see Installer script for more information.

3. Plugging in 680x0 modules

Owners of 68020-68060 systems should copy the turbo versions of some of the modules over the 68000 installation. Simply type "Copy libs/68030 to LIBS: all" or similar.

4. Plugging in PackerSupport

PackerSupport/Unpack-SVObject/libs/svobjects does contain the support module for XPK and PowerPacker Data (PP20). Optionally copy this to LIBS:svobjects and the config from env to ENVARC: (and ENV: as well, of course)

5. Plugging in SQ-Operators

Steve Quartly's great additional svoperators can be found within "SQ-Operators/". Just install those separately, as already done for PackerSupport before.

6. Using VMEM: (Needed)

 Additionally, you should create an assignment called "VMEM:", which is intended to contain virtual memory image files, which may be created either by superviewsupport.library (vmem_XXXXXXXX), JPEG.svobject (jpeg_XXX) or other modules.
 It does not always make sense, to place this directory somewhere at a Ram-Disk, because most programs will perhaps only try to place files there, after they already checked, that there's no more memory available (so even your Ram-Disk would not grant more space).

It is suggested to place this Assign on a fast harddisk or partition, with a high AddBuffers value.

7. Plugging in Ghostscript/Postscript (TM)

If you have Ghostscript installed, you may want to use it for reading Postscript (TM) files - if not, you may consider to install it for such uses.

See

GhostScript.svobject
 description for more information.

C. To Be Checked Also

=====

1. Datatypes-Support

"datatypes.library" V39+ is only needed, if OS3-DataTypes are wished to be supported - not available for OS2 users.

2. Other Libraries

Please take care, which libraries are also additionally needed by the single SVOjects/Drivers/SVOperators.

3. Additional Settings

Directory Caching:

~~~~~

Buffer values (set via C:AddBuffers) for the used Drives should at least be about 25, sometimes more may be quite useful.

Stacksize:

~~~~~

If you don't set this one to a reasonable value, the system might crash when using some specific modules (e.g. Dither24Bit). It is strongly recommended to use values >= 32768 bytes.

4. Preferences

For modifying any specific ControlPad settings, you should use SVPrefs (optionally copied to SYS:Prefs during Installer-based installation).

Nevertheless sometimes modules do not come with the required .cpinfo files, so it would become necessary to edit ControlPad

files directly at their locations ENV:SuperView-Library and ENVARC:SuperView-Library, maybe by using the descriptions inside the doc-file "ControlPads".

5. LZW License Check

If LZW usage is license-free in your country type the following two commands in your Shell:

```
SetEnv ENV:SuperView-Library/LEGAL.controlpad LZW=ENABLED
SetEnv ENVARC:SuperView-Library/LEGAL.controlpad LZW=ENABLED
```

otherwise

```
SetEnv ENV:SuperView-Library/LEGAL.controlpad LZW=DISABLED
SetEnv ENVARC:SuperView-Library/LEGAL.controlpad LZW=DISABLED
```

If you don't correctly answer this, you hereby agree in paying any possible license fees by yourself.

You may compare this with a freely distributable LZW source code that you decide to compile and use - by respecting the copyrights on it.

1.7 Overview of currently available SVOjects

Available SVOjects (as far as included) :

SVOject	Type	Read-Support	Write-Support
AmiFIG (fig)	EXT	(see AmiFIG docs)	(not available)
BMP (Win,OS/2)/RIFF	EXT	max 8/24bit Files	max 8/24bit Files
C64 (Koala, Doodle)	EXT	max 4bit Files	(not yet available)
CDR (Corel Preview)	EXT	(various)	(not available)
EPS(F)-Header	EXT	(various)	only from 24bit sources
FastILBM24	EXT	24bit Files	(-> via IFF-ILBM)
FBM (*nix)	EXT	max 8/24bit Files	max 8/24bit Files
Ghostscript	EXT	(see GS docs)	(not available)

GIF upto 89a		
EXT	max 8bit Files	max 8bit Files
Icon (W*nd*ws)		
EXT	16 Color Icons	(not available)
IFF-ACBM		
EXT	max 8bit Files	max 8bit Files
IFF-DEEP		
EXT	24bit Files	only 24bit Files
IFF-ILBM		
EXT	max 8/24bit Files	max 8/24bit Files
IFF-YUVN (misc)		
EXT	24bit Files	only 24bit Files
IMG (GEM/V*nt*ra)		
EXT	max 8bit Files	(not yet available)
JPEG (JFIF)		
EXT	max 8/24bit Files	always as 24bit Files
MAC (MacPaint)		
EXT	max 1bit Files	(not yet available)
PCD (PhotoCD)		
EXT	24bit Files	(not available)
PCX upto V3.0		
EXT	max. 8/24bit Files	max. 8/24bit Files
PICT-2 (Mac)		
EXT	always as 24bit Files	(not available)
Pictor/PCPaint/PIC		
EXT	max 8bit Files	(not yet available)
PNG (Network Gfx)		
EXT	max 8/24bit Files	(not yet available)
PNM (*nix)		
EXT	max 8/24bit Files	always as 24bit Files
QRT/POV		
EXT	24bit Files	24bit Files
SGI		
EXT	max 8/24bit Files	only 24bit Files
SunRaster (RAS)		
EXT	max 8/24bit Files	max 8/24bit Files
SVG Graphics		
EXT	max 8/24bit Files	max 8/24bit Files

```

Targa
          EXT    max 8-24bit Files    max 8/24bit Files

TIFF (V5.0)
          EXT    max 8/24bit Files    max 8/24bit Files

Utah RLE
          EXT    24bit Files          24bit Files

WPG (WP Draw BitMap)
          EXT    max 8bit Files        (not yet available)

OS3-DataTypes
          INT    all Pictures/DTs      (not available)

picture.datatype V43
          EXT    all Pictures/DTs      (not available)

```

There's also an Unpack.svobject in the PackerSupport-Directory, which allows reading of packed files without explicitly unpacking these before (supports XPK and PP20).

It is as well possible to save graphics as C-Sourcecode
 .

Planned SVOjects (no guarantee !) :

- MTV
- Imagine Textures
 (if possible)
- RGB8/RGB24
- Atari formats
 (if I get my hands on some pictures ->
 send
 some, if PD !)
- and more
 (you may
 send
 me gfx's and/or descriptions, if PD)

Do not send any graphics via email.

1.8 Overview of currently available SVDdrivers

Available SVDdrivers (as far as included) :

SVDriver	Requirements	BITPLANE	ONEPLANE
ECS			

	ECS*, OS V2.04+ (V37)	(system)	8/24bit
AGA	AGA*, OS V3.x+ (V39)	(system)	8/24bit
CyberGraphX			
	CyberGraphX RTG System	8bit	8/24bit
EGS-System			
	EGS-Graphic-Libraries V7	8bit	8/24bit
Merlin			
	Merlin Graphics Card	8bit	8/24bit
OpalVision			
	OpalVision Card	8bit	8/24bit
Picasso II			
	PicassoII-Card	8bit	8/24bit
Retina			
	Retina Z2/Z3 Card	8bit	8/24bit

Additional or alternative NOT INCLUDED SVDdrivers (e.g. on AmiNet)

Picasso			
	PicassoII-Card	8bit	8/24bit

* ECS- and AGA.svdriver will both work with Graphic Cards, if they are configured with "BITMAPCOPY=RTG" and the GfxCard software does supply a Workbench Emulation and/or ScreenPromoter. If you own a Graphics Card, without having an appropriate SVDriver for it, always try using AGA.svdriver at first.

Planned SVDdrivers:

"Send me a card, I write the Driver !"
(at least trying to do so, then... ;-)

1.9 Overview of currently available SVOperators

Available SVOperators (as far as known and/or included) :

SVOperator

24BitToHAM

dithers 24 Bit RGB to HAM6/HAM8

AnyTo24Bit

converts input to 24 Bit

CallPNM
Call preset external PNM operator

Crop
"crops" boxes of any size from 8/24 Bit graphics

Dither24Bit
dithers 24 Bit RGB to 2..256 Colors

ExtractBlue
extracts Blue values from (upto) 256 Colors or 24 Bit

ExtractGrayScales
converts (upto) 256 Colors or 24 Bit to Gray

ExtractGreen
extracts Green values from (upto) 256 Colors or 24 Bit

ExtractRed
extracts Red values from (upto) 256 Colors or 24 Bit

HilbertDither256
dithers to Black & White

LeftToRight
turns left to right ...

OptimizePalette
reduces depth by filtering unused or redundant colors

PaletteDither
dithers (optionally in parts) to a user-supplied palette

Rotate
rotates by 90/180/270 degrees ...

RotateFree
rotates freely by 0..360 degrees about a given point

Scale50
scales to half/double size

TopToBottom
turns top to bottom ...

XOR
nice effects ...

Just for information:

- this distribution also includes a bunch of image processing operators by

Steve Quartly
(SQOperators) like

SQBentley.svoperator

SQBlur.svoperator

```

SQContrast.svoperator      SQDeep_Press.svoperator
SQDiffuse.svoperator      SQEmboss.svoperator
SQHighPass.svoperator     SQMosaic.svoperator
SQMotion_Blur.svoperator  SQOilPaint.svoperator
SQSharpen.svoperator      QSlicing.svoperator
SQThreshold.svoperator    SQTiling.svoperator
    
```

- buyers of the commercial program
 Picture Manager professional
 (short: PMPro) will also get some additional operators, ←
 which
 are not freely distributable (exclusively shipped with PMPro):

```

Antique.svoperator        Brightness.svoperator
Complement.svoperator     Contrast.svoperator
Convolve.svoperator       DitherFix24.svoperator
EdgeDetect.svoperator     Gamma.svoperator
Mosaic.svoperator         RGBAdjust.svoperator
ScaleFree.svoperator
    
```

(referring to Release V3.01 - may differ from version to version)

Planned SVOperators (no guarantee !) :

- some more operators for common image processing
- and more

1.10 Software supporting superview.library

- o The following programs do own a license for superview.library ←
 V11.8+ :

Program	Description	Status
SuperView	THE image viewer and converter for anyone	SW
SqOpal	THE image viewer and processor for OpalVision	FW
SuperLoader	THE loader module for OpalPaint	FW
ImageEngineer	THE image processing program for anyone	SW
PictureManager	THE image database program for professionals	COM
DRAFU	THE function plotting program	SW

- o The following programs are currently supporting superview.library upto version 11.7 :

(There may be non-commercial programs out there using it.
Also, there may occur various SVDivers, SVObjects or SVOperators as add-on's for superview.library).

- o The following programs have been written by me and do also support superview.library in some way :

Program	Description, Author	Status
SimpleView	superview.library Example Program	FW
MicroView	superview.library Example Program	FW
KFracPlus	fractal generator (saves via library)	FW
SIP	for examinations of SV-Modules	FW

1.11 superviewnote

SuperView is a program that has been designed for the purpose, to display any kind of graphic as fast and as comfortable as possible.

Fast means not only fast in reading and displaying but also in calling and using the program.

Highest possible flexibility is implemented via the external Driver-System, consisting of the superview.library, the SVObjects, SVDivers and SVOperators.

These intentions resulted in a bundle of features and options you have access to when using SuperView and installing it to your System:

- licensed "superview.library" (see directory SuperViewLibrary)
- intensive use of many special OS V2.04+ and OS V2.1+ capabilities
- support of many OS V3.x+ and AGA graphics features, as e.g. support of interleaved BitMaps
- support of all SVObjects, SVDivers and SVOperators of superview.library
- "Screen-Grabbing"
- Commodity (optional)
- ARexx-Ports (optional)
- AppIcon (optional)
- AppMenu (optional)
- AppWindow (optional)
- Clipboard reading and writing
- Localization for OS V2.1+ (not all texts yet)
- AmigaGuide OnLine-Help for OS V2.04+ (asynchroneous & localized with 2.1+)
- detailed configuration via Config-File, Cli-Options and Workbench-Tooltypes
- Graphical User Interface (GUI) for Workbench-Users (optional)
- conversion of the supported File-Formats
- information about the displayed graphics via Requester, selectable via GUI-Menu
- and more

It is Shareware and can be found on AmiNet under gfx/show (for example gfx/show/SView542.LHA).

SuperView WWW page: <http://members.aol.com/tfischb/main.htm/>

1.12 sqopalnote

SqOpal is an Image Processing, display and manipulation package especially for the OpalVision 24 Bit Graphics and FrameBuffer hardware.

It has been written by Steve Quartly and Paul Huxham and optionally uses superview.library and its svobjects and svoperators.

It is Shareware and can be found on AmiNet under gfx/show (for example gfx/show/SqOpal20.LHA).

Steve Quartly: steveq@sndcrft.DIALix.oz.au
Paul Huxham: paulh@perth.DIALix.oz.au

1.13 superloadernote

SuperLoader is a Loader Module for OpalPaint, which comes with the OpalVision 24 Bit Graphics and FrameBuffer hardware.

It has been written by Steve Quartly and Paul Huxham and uses superview.library.

It is Freeware and can be found on AmiNet in the graphics section.

Steve Quartly: steveq@sndcrft.DIALix.oz.au
Paul Huxham: paulh@perth.DIALix.oz.au

1.14 imageengineernote

Image Engineer is a shareware image processing application for any Amiga with 68020 and OS 2.x or greater.

Image Engineer can be used for tasks varying from converting images between different file formats, rendering 24 bit images down to standard Amiga screen modes, enhancing badly scanned images, applying special effects even up to advanced image composition.

What you can use it for is basically limited by what you can think of.

Registration is only 35 US dollars (45 AUS Dollars, 55 DEM).
Registration sites are in Australia (Simon Edwards)
and Germany (Andreas R. Kleinert).

Registered users will be sent a personal keyfile which will unlock all of Image Engineer removing the limits on the image size, and the 'Register Now...' requesters.

This keyfile will also work for future versions.
Keyfiles can be sent out via mail or Email.

IE can be found on AmiNet under gfx/show
(for example gfx/edit/ImEngV3.0.LHA).

Simon Edwards: sbe@yallara.cs.rmit.edu.au
IE WWW page: <http://minyos.its.rmit.edu.au/~s9407349/>

1.15 picmanagernote

Picture Manager Professional 3.x (PMPro) is an excellent commercial image catalogization program with a large number of features for creation and handling of thumbnail tables.

It is a powerful database and allows loading, processing and displaying images via superview.library and includes an automatic conversion function for catalogs of images, which allows to convert between all file formats superview.library does support ... AND MORE ...

- The GERMAN version of Jürgen Schäfer's Picture Manager can be obtained from

IrseeSoft SPCS
Meinrad-Spieß-Platz 2
D-87660 Irsee Voice: +49-(0)8341-74327
Germany Fax: +49-(0)8341-12042

Or from any distributor. The official pricing as of May 1996 has been 129 DEM (no guarantee).
Ask them for it.

- The ENGLISH version of Jürgen Schäfer's Picture Manager may be obtained from

BlitterSoft
6 Drakes Mewsm
Cronhill
Milton Keynes
MK8 OER Fax: +44-(0)1908-261488
England, UK WWW: <http://www.mag-net.co.uk/Bsoft/>

Or from any distributor. The official pricing has not been announced as of May 1996, but should be compareable to the german pricing (no guarantee).
Ask them for it.

1.16 drafunote

DRAFU plus is a function plotter, which optionally supports `superview.library` for saving the drawings from single windows.

It includes the following features:

- freely defineable function terms ($x^?$, sin, cos, tan, ...)
(function parser by Dipl.Ing. Ulrich Degens)
- unlimited number of sizeable windows on the screen
- unlimited number of functions per window
(presented as a list)
- upto 256 colors on the screen (AGA):
 - freely selectable draw color for each function
 - freely selectable background, drawing and axis color
for each window
- loading and saving of function lists
- flexible axis' description (decimal, logarithmically, scientific)
- single/double axis description
- saving windows drawings in any of the graphics file formats
`superview.library` does support (optionally, library is not needed)
- hardcopy function

AND MUCH MORE

DRAFU can be found on AmiNet under `misc/math`
(for example `misc/math/DRAFU102.LHA`).

Author: me ;-)

1.17 Thanks, greetings, credits and the rest

I perhaps have to thank many persons for supporting me with ideas,
Bug-reports and so on :

Thanks go to (in alphabetical order) :

* Jan van den Baard

... for his great tool `GadToolsBox`, which I formerly used to
design the GUI of `SVPrefs`.

* Ian O'Connor

... for the `Designer`, which I now do use for GUI creation.

* Simon "ImageEngineer" Edwards

... for a lot of things concerning `superview.library` :-)

* Gerd Frank

... for Beta-Testing, Bug-Reports and for his many ideas and suggestions,
concerning `superview.library` and `SuperView` (also see notes there) !
And last not least for the first draft of the Guide Documentation... ;-)

* Roman Patzner

... for the nice icons he designed for use with Martin Huttenloher's MagicWB (8 Colors minimum) and sent to me for inclusion with SuperView. With superview.library only the InstallerScript-Icon is used yet (several times ;-)

* Steve Quartly and Paul Huxham

... for the various versions of OPAL.svdriver included with the library (Steve and Paul) and for Steve's SQ-Operators, included with his friendly permission. Steve also did "SuperLoader" and "SqOpal" together with Paul Huxham. THANKS !

* Jürgen Schäfer

... for specific Beta-Testing and related Bug-Reports, as well as several useful hints and suggestions on implementing new features to the library

* Martin Schulze

... for uploading SuperView onto the AmiNet and including it into the SaarAG series, so that it reached more people out there. Also for sending and receiving all those mails, which did not find their way through the labyrinth of Fido-Gateways ;-)

* Detlef Winkler

... for the new Icons for Doc- and AmigaGuide-Files, included with the Library. He also designed some graphics and icons for use with SuperView and new drawer icons. Additionally, he had a lot of ideas, suggestions and graphics for 24 Bit IFF-ILBM, IFF-YUVN, IFF-DEEP and SGI support.

* to the translators

Italiano: Alessandro Basso (cralex@amiga.dei.unipd.it)
Luca Giolo (grifon@vega.unive.it)
Svenska: Patrik M Nydensten (nyden@algonet.se)
Torbjörn Aronsson (torbjorn.aronsson@mailbox.swipnet.se)

* and last NOT LEAST

- all _registered_ users of SUPERVIEW for supporting Shareware !!
(ORDER YOUR KEYFILE NOW !)

- the people mentioned below (still in alphabetical order ;-):

Joachim	Baumeister	supplier of my second UseNet access
André	Bergmann	made bug-reports and uses superview.library
Alex	Carbin	TuC / Co-Sysop Century
Rüdiger	Dombrowski	ADX-Datentechnik, "Amiga Demos+Tools" CD-ROMs
Thomas	Dorn	Author of XiPaint (thx for 3.2 and 4.0)
Sven	Drieling	"Indy" - Creator of "Power-Brei" DiskMag and the Oberon2-Language interfaces and modules.

Oliver	Eichhorn	author of EGSTasy
Thomas	Eigentler	Programmer of MERLIN.svdriver (included)
Thomas	Fischbach	my current Fido-Boss and supplier of "GM-Box" SupportBBS
Fred	Fish	AmigaLibDisks and Fish CD-ROMs
Stefan	Grad	GPD-Disks
Klaus	Holtorf	for several information on graphic file formats
Ing.-büro	Helfrich	for supplying the PiccoloSD64 Graphics Card
Stefan	Kremer	TuC / Sysop Century
Alex	Lange	Time PD-Disks (aps-electronic)
H.P.	Lattka	Franz PD-Disks
Andreas	Manewaldt	Taifun PD-Disks
Axel	Melzener	Game Object Design
Andreas	Neumann	Creator of the PCQPascal-Language includes and modules
Patrick	Ohly	Programmer of PICASSO.svdriver (AmiNet)
Michael	Petrikowski	Amiga Szene PD-Disks, SEP - Software-Entwicklung
Albi	Rebmann	supplier of my ftn.sub.org UseNet domain/address
Michael	Trautmann	PC-Programmer, for information on Win icons
Christian	Wincziers	Sent me some PICT-2 graphics. Thanks !
Florian	Zeiler	Sending me the RetinaZ3 and PicassoII graphic cards enabled me to write own Drivers for these.

- some users, which e.g. reported bugs via mail, e-mail or telephone or did something else related to my programming work on the library (only the ones, which have not already been listed above) :

Thomas	Alexnat	
Rudi	Brand	
Aaron	Digulla	
Michael	Flad	
Grant	Fribbens	
Thomas	Gundlach	
Richard	Hartmann	
Michael	Kilimann	
Mika	Lundell	
Mats	Jansson	
Jim & Becky	Maciorowski	(thanks for the nice card :-)
Neil	Mohr	
Andy	Philpotts	
Fabio	Rotondo	
Jürgen	Schneider	
Tommy	Simonsen	
Klaus	Stengel	
Reinhard	Theling	
Henrik	Tikanvaara	
Marco	Vernaglione	
Fabien	Wernly	
Tilo	Winkler	(thanks for the strange TIFFs ;-)

and more.

Maybe I forgot somebody to list here, but nobody's perfect... ;-)

Please note: If you wrote to me and didn't get an answer,
this need not necessarily mean, that I didn't

answer - mails can get lost sometimes.
Please tell me and ask again!

1.18 How to contact the author

```

|   You may reach me the following way.
|   Send bug-reports, money or whatever to:
|-----|
|   * SuperView Development & Registration *
|   * DRAFU Development & Registration *
|   * Image Engineer Registration Site Europe *
|
|
|           Persistant Software
|
|           Andreas R. Kleinert
|           Sandstrasse 1
|           D-57072 Siegen
|           Germany, Europe
|
| Any snail mail to the old address will still be routed.
|
|           Phone:  +49-271-22869 also FAX + AM
|                   +49-271-22838
|
|           Weekdays after 17.00h.
|
|           When calling via phone you may leave a message,
|           if I'm not available - but don't expect me
|           calling back to USA, Australia, ... since
|           german phone rates are HIGHLY expensive.
|-----|

```

SuperView WWW page: <http://members.aol.com/tfischb/main.htm/>

E-Mail:

DO not SEND ANY binaries (or uuencoded) VIA THE
FOLLOWING EMAIL ADDRESSES, EXCEPT MAYBE small ONES
VIA t-online.de (smaller or equal 16 KB).
THANK YOU.

```

- Fido      Andreas Kleinert 2:2457/350.18
- Usenet
              Andreas_Kleinert@superview.ftn.sub.org  (Fido-Gate)
              Andreas_Kleinert@t-online.de           (T-Online)
              AK@COB.wwbnet.de                       (Z-Netz)

```

- If nothing else works, try one of these public
Fido-Usenet gateways:

In Germany:
Andreas_Kleinert@p18.f350.n2457.z2.fido.sub.org

From USA or elsewhere:

Andreas_Kleinert@p18.f350.n2457.z2.fidonet.org

Please note, that the "superview.ftn.sub.org" domain will perhaps be renamed in late 1996 (superview.*).

When reporting any bugs, please don't forget to include a detailed description of the bug and tell me, if it is reproduceable or not. Also mention the version number of superview.library (and e.g. SuperView) which caused the bug and describe your system configuration (Amiga model CDTV/CD-32/600 (HD)/500 (+)/1000/1500/1200 (+/HD)/2000/2200/2500 (UX/AT)/3000 (+/T)/4000 (/030/040/060/PPC) (T)/DraCo (/040/060), Kickstart/OS Version, RAM, HardDisk, GfxCard, any special Hardware/Software)

Since there recently have been some problems: please don't forget to mention, whether your machine is accelerated in any way, or not (68020/.../060 plus 68881/882, PMMU or PowerUp/PPC).

1.19 The future of superview.library...

The Future (or: The Undiscovered Zone ;-)

=====

Well, ideas cannot be planned and creativity is not able to be calculated ;-)

So I can only express, what I'm thinking about the future of the library: There are many things, which I want to implement, on the other hand there are things, which are just necessary to be implemented sooner or later. So much work has to be done (still).

I'll try to do this, but I hope that there's enough support from the Amiga programmers all over the world (via Inter/AmiNet, Fido, etc.) to help me to continue doing so.

This does not only mean financial support, but also additional programming efforts like third-party improvements on the library (why not also write your own SVOBJECTS/SVDRIVERS/SVOPERATORS based on the last official freely available Programmers-Sources as been released with V11.7 of the library ?)

- Andy

1.20 Known bugs, crashes and their possible workarounds

Harddisk MaxTransfer problem	(workaround possible)
Startup crashes - processor dependent	(hopefully fixed)
Problems with specific applications	(workaround possible)

1.21 Harddisk MaxTransfer Problem

IDE-MaxTransfer Problem

Since superview.library usually holds very large buffers within memory, it also likes to read and write these completely from and to disk. This means, that the specific device drivers are confronted with quite large values of bytes to be read or written, which perhaps usually does not happen very often.

Sometimes the firmware of IDE-Harddrives, like shipped with the A4000/030-040 or A1200HD, does not support transfers of blocks larger than 64K (65535 Bytes) during one single write operation. Ususally the DOS splits larger writing calls to take care of this restriction. But since this is just a lack of performance and actually does not comply to the IDE/AT standard, the default value for this "MaxTransfer" is not 0xFFFF (64K) but 0FFFFFFF or 0FFFFFFFF instead.

If any written graphics files are mysteriously damaged or will be read incorrectly (writing is more critical than reading), you should start your "HDToolBox" and select "Partition Drive" for the concerned HardDrive. After that activate "Advanced Options" and chose "Change". Modify the "MaxTransfer" field, so that it does reflect "0xFFFF" then. After that leave all the windows by confirming "OK" and select "Save Changes to Drive" (no longer disabled) on the first window.

```
>>> Do not change any other settings within "Partition Drive", if <<<
>>> you don't know, what you're doing, since actually partitioning <<<
>>> your HardDisk would cause your complete data to be lost. <<<
>>> If you did change something you didn't want to change, just <<<
>>> "Cancel" the whole thing and start from the begining. <<<
```

Perhaps you'd like to know, why I did mention this here ?!
Well, some weeks after I bought a new M*xt*r 540 MB HD, superview.library did seem to have x-time more bugs than before, which almost all could not be explained rationally: writing a "clean" buffer to disk in several file formats (did not matter), with this buffer still being valid after the write operation, resulted in damaged graphics after loading. If uncompressed, the data still was all there, but like a kind of mosaic, with always some blocks at the wrong places...

It took me some weeks to actually realize the bug itself and approximately two days to find out, *why* it happened... %-(

1.22 Startup Crashes - Processor Dependent

The bugs described in the following should hopefully already be obsolete as with superview.library V12.5 and/or V12.7.
If not, please report it to me!

Running SuperView-Library on systems with 68060 processor (and others)

There had been a few reports about problems with getting the library started on 68060 powered systems.

According to Michael Kilimann (mike@kily.ruhr.de) and others, those may also concern some 68040 systems sometimes.

It seems to me, as if either only some people got these difficulties, or that these first appeared after the V4.63 release, thus being quite new bugs.

Namely, people mentioned Draco/060 and CyberStorm060 systems.

At this place thanks to Mats Jansson, which showed up a way to temporarily bypass this problem by removing the following of superview-Library's modules. When deleting

```
JPEG.svobject
PICT.svobject
and   SVG.svobject
```

from LIBS:svobjects, the Library boots up as usually (with former versions SnoopDOS sometimes reported, that the library did freeze and crash the system when opening one of the mathieeedoub-Libraries. In such a case sometimes deactivating SnoopDOS did help - although it perhaps was not the fault of SnoopDOS).

Please also delete SVO.svobject, which is no longer included and therefore had not been fixed (and overwritten by never versions, then).

Will try to fix the bugs within the rest of the concerned modules as soon as possible.

When owning an 68060 and using the library, you should try the following:

- install it as usually and try, whether it runs
- if it does not run, please check, whether Installer actually did install the 68030 versions of several modules, or whether it did recognize the 68060 as a plain 68000 only (may fix JPEG problems)
- if it still does crash then, try removing the mentioned modules from LIBS:svobjects - not all together, but instead one after another (and testing after each removal)

Since I don't own a 68060 at the moment, this are the only hints for me to give you, but this should suffice, until the bugs finally have been fixed (BTW, for SVG and JPEG there are Datatypes available, and PICT2 is quite exotic).

1.23 Problems with specific applications

Here's an alphabetical list of programs, which either cause superview.library to crash, or which do crash relatively

often and unexpected (so that it might seem, as if superview.library crashed):

- AutoCLI V2.xx © by Nic Wilson

Problem:

AutoCLI sometimes might crash, when there's no appropriate window available when e.g. trying to open a shell via Hotkey.

Solution:

(don't use the hotkey feature)

- EGS libraries V6 and V7 © VIONA Development

Problem:

When flushing the EGS libraries (at least after using the Amiga emulation mode), it seems that the libraries will cause recoverable alerts with OS 3.x. Maybe on some systems crashes will occur. Don't know, whether the libraries are really the source, but it's likely.

Solution:

(don't flush)

- LX/020 1.03 © 1993 Xenomiga Technologie (Jonathan Forbes)

Seems to have problems in extracting larger archives of LHA.

- NewMode V3.3 (and below) © 1992-95 by Andreas Linnemann

Problem:

Has been reported to cause serious problems when running together with e.g. SuperView (when attaching a fixed ViewMode to the program).

Solution:

(Already fixed for newer versions.

I, personally, would strongly recommend not to promote any SuperView Screens via NewMode, since now for almost any configuration there should be suitable SVDivers - alternatively also a RTG-configuration for AGA.svdriver)

- SuperDark 1.5c,V2.1a © by Thomas Landspurg

Problem:

Some of the Screen Blanker modules will crash on AGA OS3.x systems.

Solution:

Find out, which of them do so (e.g. ASwarm, as far as I know) and deactivate them !

- VMM

Problem:

Someone told me, that VMM caused SuperView(-Library) crashing because of Enforcer Hits, which did not happen without it.

Well, consider this facts:

superview.library usually does allocate all memory with the MEMF_PUBLIC flag set. This memory will not be "virtualized" by VMM because there's a common agreement - not respected by many programs - that this is memory which has to be in RAM always. This default behaviour may be overridden by setting special settings within VMM, e.g. setting 10240 for both - MEMF_PUBLIC as well set or cleared - within "advanced options".

By doing so, VMM might be caused to not only virtualize large graphics buffers, but maybe also Messages, IORequests and other things. This simply may cause crashes.

Solution:

(If it does not work, don't use it with SuperView - I've been told that the above method works without problems, but you try it at your own risk - maybe the library's behaviour will change in the future. Either better using the MEMF_PUBLIC flag or crashing by being badly virtualized ;-)

1.24 History

Please note the version-dependencies :

superview.library	SVObjects	SVDrivers	SVOperators
Version 1	-	-	-
Version 2	Version 1	-	-
Version 3-8	Version 1-2	Version 1	-
Version 9-10	Version 1-2	Version 1	Version 1
Version 11	Version 1-2	Version 1-2	Version 1
Version 12-13	Version 1-3	Version 1-2	Version 1-3

V13.2 (18.6.1996) :

-
- LIBRARY:
 - rewrote startup-code: it was already really safe before, but now it has been switched from 100% asm to mixed Asm/C (that rocksolid one). Also changed order of initialization modules. Smarter now. Saved some 100 bytes ;-)
 - NEW SVObjects:
 - CorelDraw (TM) CDR Previews:
 - CDR.svobject reads preview pictures of of CorelDraw .CDR files
 - Ghostscript:
 - GhostScript.svobject
 - allows importing
 - Postscript (TM) vector graphics as bitmap using an available Ghostscript installation. PDF is also basically supported.
 - AmiFIG (fig):
 - AmiFIG.svobject uses an installed "fig2dev" (from the AmiFIG port) to create Postscript (TM)

files from FIG vector drawings - which then can be read by GhostScript.svobject.
Thus you'll need both to be installed and configured.

- SVOjects:
 - BMP: BMP.svobject now also reads RIFF-BMP (DIB)
 - EPS:
 - now optionally allows extraction of Postscript (TM) part of EPS files, which requires Ghostscript and GhostScript.svobject being correctly installed and configured
 - also writes Postscript (TM) now in form of EPS files without trailers (no TIFF) from 24 Bit sources
 - GIF, PICT:
 - fixed mem trash bug (4/8 bytes)
 - GIF:
 - now allows writing of transparent GIFs (possible with GIF89a only); useful for creation of HTML pages (-> Patrik Nydensten, Sven Drieling)
 - ILBM: (see there)
 - ACBM, ILBM, DEEP, YUVN:
 - fixed chunk recognition (may have read beyond ID array size)
 - changed/fixed Clipboard file recognition
 - PICT, QRT: (see there)
 - PNG: (see there)
 - DEEP, SGI, SunRaster, Targa, WPG, MAC, FBM, IMG, C64, PNM, Pictor, WinIcon:
 - added new, rock-solid startup code
 - DEEP, SGI, SunRaster, ILBM, C-Source, ACBM, Targa, EPS, YUVN, QRT, FastILBM24, PCD, (UtahrLE):
 - fixed bug in initialization code (may have caused crashes when flushing/reloading)
- SVOperators:
 - CallPNM: (see there)
- Localization:
 - updated italian "superviewlib.catalog"
 - Thanks to Alessandro Basso for that one !!
- OnlineSupport:
 - SKYNET offline as with 31 May 1996!
 - GM-Box offline as with 31 June 1996!
 - removed reference to n2usx.sauerland.de domain
 - NEW Support BBS and Z-Netz-EMail: CHAOS-Box (domain COB.wwbnet.de)
 - new Fido address (mail to old address will be redirected to the new one)
- Documentation:
 - updated information where necessary
 - added some important information to

"Requirements" section, so that the needed configuration efforts for Ghostscript, AmiFIG, picture.datatype V43 and so on becomes more obviously.

V13.1 (16.4.1996) :

-
- LIBRARY:
 - added the concept of "exclusion" of single modules from the startup initialization of the library by user's choice. SVOobjects, SVDdrivers and/or SVOoperators not to be loaded each time can be "excluded" by adding their names to specific configuration files. May increase speed on startup, save memory and reduce the need to call "avail flush" or actually delete modules from their Libs subdirectories (see documentation of "SVPrefs").
 - Has some disadvantages (excluded modules won't be recognized until again - maybe temporarily - loaded by hand or re-included), but these don't make anything worse, since this feature can most simply be ignored.
 - Preferences:
 - added GUI support for "exclusion" model of modules
 - NEW SVOobjects:
 - C-Source.svobject creates C-Source code in chunky/RGB notation from any graphics (<= 8 Bit and HAM as chunky, 24 Bit as RGB)
 - SVOobjects:
 - GIF, TIFF:
 - added LZW license check to GIF and LZW-TIFF
 - ILBM:
 - now finally, ultimatively, definitely, absolutely fixed ILBM buffersize calculation stuff (simply had been blind - thanks to -> Simon Edwards)
 - fixed mungwall hits when writing 24 Bit IFF-ILBMs (-> Simon Edwards)
 - SVSupport-Lib:
 - fixed grabbing of interlaved screens with ScreenToBitPlane() (-> Simon Edwards)
 - Programmers:
 - fixed superview.fd: GetTypeExtension (-> Simon Edwards)
 - added new function SVL_SetDTMode(), which allows enabling/disabling datatypes support (if enabled, it also supersedes the LIBRARY controlpad entry "ANYDATATYPES" by specifying own settings)
 - Documentation:
 - updated Simon's EMail address
 - OnlineSupport:
 - SNOWBALL is offline!
 - gm-box/mediaserve domain canceled
 - skynet domain changed
 - new WWW page and FTP address with SuperView-Support (GM-Box)
-

- added other doc files as external links
- rewrote SQ-Operators.guide
- added more links to this document and updated information where necessary

V12.9 (14.3.1996) :

- Misc:
 - PaletteDither had been missing due to a fault in an internal batchfile. Now included. I'm sorry.
 - new, additional FTP support (see chapter)
- SVOjects:
 - JPEG: - fixed a heap of bugs (-> Mika Lundell, Simon Edwards)
 - ILBM: - fixed another heap of bugs (-> Simon Edwards, Jürgen Seligmann via Florian Zeiler)
 - TIFF: - fixed reading of intel style 24 Bit (-> Lothar Köster via Florian Zeiler)
 - PCX: - misc
- SVOperators:
 - included updated SQ-Operators from Steve Quartly (with progress indicator support, etc.)
Thanks, Steve!
(fixed a small bug in my installer script: when installing these SVOperators, their CPInfo files would not have been placed in the correct ENV path. Fixed.)
- SupportLibrary:
 - caused crashes when saving from Screens under some circumstances (e.g. from ImageEngineer or SuperView) (-> Simon Edwards)
 - misc (ECS "Cannot handle HAM data" problem)
- Documentation and Support:
 - new support Mailbox with mail and binary areas: SNOWBALL
 - completely rewrote and updated SupportBBS text (now named "OnlineSupportSites")
 - reworked and updated documentation
 - added Sven's new Fido address to Power-Brei.txt

V12.8 (10.2.1996) :

- new SVOjects:
 - CyberDataType.svobject

Supports (CyberGraphX') picture.datatype V43, which upgrades DataTypes to 24 Bit handling.

When CyberGraphX and picture.datatype V43+ are installed in your system, this module will supersede the internal DataType svobject (for graphics only) and thus allow extraction of 8 Bit Chunky and 24 Bit RGB graphics from datatypes.

You need special datatypes for that (as coming with picture.datatype V43).

 - PNG.svobject (Portable Network Graphics)

Allows reading of 8, 16, 24, 32 Bit PNGs.

- new SVOperators:

PaletteDither is an operator, which allows dithering to a fixed palette, where you may freely define the palette by specifying any (upto 256 color) GRAPHICS as a source, of which a specified number of colors will be taken from its palette and the rest will be freely set by the dithering routines - without redundancies.

So you may:

- specify an already dithered graphics to which a number of other graphics should be equally in parts of its palette or the full palette
- simply specify a 1x1 pixel brush IFF file as a colormap to be used fully or in parts (instead of an IFF-ILBM file with only a CMAP in it)

For what is that needed ?

- for creation of ANIMations, where either all pictures must share the same palette, or only a specific number of colors may be changed
- for creation of games and applications, where 24 Bit graphics of many sources should be placed on one single 256 color screen

You can specify the depth the 24 Bit input should be dithered to and also how many colors should be predefined by the given palette (which may be larger then).

This functionality has only been native to expensive image processing applications yet.

- SVOjects:
 - BMP:
 - fixed reading of several non-standard BMPs (e.g. from OS/2 or newer "Windozes")
 - added read support for RLE4 compression
 - ACBM,
 - ILBM:
 - now filtering unsupported bit depths instead of letting the machine crash...
 - YUVN:
 - now is able to distinguish between YUV gray (400), 411, 422, 444, 211 and 222 when reading (211 and 411 have been tested).
 - fixed reading and writing of 411
 - maybe first useable version (in/output of the others would only have been nearly correct when being converted to gray)
 - QRT:
 - misc (see there)
 - TIFF:
 - misc (see there)
 - JPEG:
 - linked with wrong math stubs. Caused crashes (-> Mika Lundell)
- SVDrivers:
 - CyberGraphics: (see there)
- Docs:
 - added note on LX/020 problems when extracting archives

- rewrote sections about email addresses, known bugs, and more
- added LZW/Unisys note to Copyright section
- included NotesAndHints.guide to main guide-file and left out unimportant information

V12.7 (13.1.1996) :

- CRASHES IN GENERAL (math-related)

It seems as if the opening (can't say, whether the libraries themselves) of transient math libraries (like "mathtrans.library" together with "mathffp.library" or "mathieeedoubtrans.library" together with "mathieeee.library") did cause all the concerned SVOjects (recently JPEG, but no longer) and/or SVOperators (RotateFree) to crash on certain systems like 68060, some 68000 and some 68040 (on my system only, when running SnoopDOS at the same time to catch library openings).

May be a problem of shared libraries and ramlib in conjunction with specific math libraries, a specific SAS/C bug
- or just accidental...

No problems when only using "mathffp.library" or using link libraries for both, non-transient and transient functions.
No more crashes at startup (Guru #4).

We now do that this way. It works. Really.
No more tries...

- SVDdrivers: CyberGraphics

the CyBERGraphX driver did distort any 8 Bit odd width graphics (not divideable by 16 without rest) when displaying under OS V3.00 (V39).
This did not happen with graphics < 320x240 and the SMALLSCREENS option unset and it did not concern OS V3.1 (V40) users.
(-> Jürgen Schäfer)

- SVOperators: OptimizePalette, RotateFree

both had bugs or did crash

- Docs: - several changes

V12.6 (22.12.1995) :

- several Modules: - PCX, GIF, PICT, SVG, UnPack, ECS, ExtractGrayScales, RotateFree, ...:
second revision of startup-code with correct initializers (for e.g. lib_Revision)
- SVDdrivers: - ECS:

Usage of V39 options caused odd-address errors under V38 and below.

With V2.7 of ECS.svdriver displaying under V38

- and below perhaps was impossible. Fixed.
 (-> reported by Klaus Schneider and Sven Drieling)
- SVOjects: - GIF: fixed handling (skipping) of non-graphics GIFs
 - PCX: fixed handling of planar graphics with a width > 2048
 - JPEG: did again reference JPEGTMP:
 (-> Sven Drieling)
 - Support-Library: - now does no longer drop comments and linefeeds when loading/saving
 (-> Steve Quartly)
 See includes and docs for more information.
 - Programmers: - added SVOject, SVDriver, SVOperator flag to indicate "break while still in progress" support (Includes). Not set anywhere yet.
 (-> Simon Edwards)
 - Docs: - added "68060-Users" and "IMPORTANT.txt" to main library-guide file's new section "Known bugs and workarounds", which quite better structured and accessible than before
 - added note on how to get the OpalVision driver working in cases with black screen or distorted display (color 0 blue component adjustment)
 - reworked documentation

V12.5 (16.12.1995) :

-
- MAJOR bugfix release for 68060, some 68040 and others
 - should no longer crash at startup time
 (and perhaps also not later ;-)
 - several Modules: JPEG, PICT, Unpack, SVG
 - added new, rock-solid startup code
 - replaced IEEE with FFP support (when used), due to former crashes
 - hopefully no more crashes on certain systems (especially 2000/4000 with 68000 and 68060)
 -> users of PMPro modules would perhaps also need an update of those
 - SVOjects: - JPEG

Completely rewrote JPEG - now supports progressive JPEGs, many settings to be done for faster, more/less precise or qualitative decoding, and so on.
 Basing on the IJG sources V6.
 Now fully reentrant for parallel uses within the Amiga's multitasking system.

FASTER.

Now allows several different methods to influence the way JPEG data is read and written, decompressed and compressed, decoded and encoded.

MAJOR rework for professional JPEG usage.

Hopefully also fixes some of the 68000 and 68060

problems -> please send reports !

- PICT:

- changed some compiler options and tried to prevent 040/060 crashes
- and more

- SVG:

- reduced dependence from xpk-library, which also should help avoiding the 040/060 crashes (since no longer opening those library during initialization, but maybe later)
- and more

- LIBRARY: - recompiled with new GST

V12.4 (24.11.1995) :

- SVOjects: - JPEG (misc)
- MISC: - various ;-)
- Prefs: - SVPrefs now cleaned from Ehits and using Designer GUI creation system for better V39 functionality and font-sensitivity
- Library: - added italian catalogue for the library, by Alessandro Basso
- Docs: - added "68060-Users" text with some workaround-hints originally by Mats Jansson for getting the library running on 68060 systems
- removed "HowToInstall.doc" and instead rewrote and updated the corresponding sections within the library guide file

V12.3 (29.10.1995) :

- SVOjects: - OS3 DataTypes: - now displays (maybe shortened) Author, Anno and Copyright information in Info Requester
- modified way of attaching DataTypes to Screens and extracting BitMaps from DTs
- faster closing
- JPEG: hopefully fixed last 68000 problems
- ILBM: fixed for loading masked images
- SVDdrivers: - ECS, AGA, CyberGraphics: faster closing
- SVOperators: - 24BitToHAM: enhanced ViewMode "generation"; tries to keep old one whenever possible
- NEW SVOperators: - RotateFree (by any angle, about any point)
- Library: - fixed some possible Enforcer Hits
- MISC: - change "Locale" directory organization
- Documentation: - updated "MoreProjects" text

V12.2 (14.10.1995) :

- SVOjects: fixed/changed TIFF

- removed all 68040 versions, since there's no actual difference to the 68020/030 versions (in the future there may appear floating point versions, specifically for the 040/060, but with integer versions there's no effect as with SAS/C). Saves a lot of space.
- added Steve Quartly's SQOperators: many thanks to Steve at this place !!
- Documentation: the new phonenumber was wrong
- Installer-Script: - had a bug in 68030+ version installation
 - now also asks for installing add-ons like PackerSupport and SQOperators (-> Klaus Stengel, various)

V12.1 (10.10.1995) :

-
- introduced unlinkable GfxBuffers, default file extensions and progress indicators:
 - reworked all SVOperators for unlinkable GfxBuffers and progress indicators (-> Simon Edwards)
 - reworked all SVOjects for unlinkable GfxBuffers, default file extensions and progress indicators (-> Simon Edwards, Steve Quartly, Jürgen Schäfer, Sven Drieling, several)
 - upgraded SVDriver specification for progress indicators, also (not yet widely supported)
 - added basic possibility to break work-in-progress-operations via progress indicator function (e.g. when a user presses ESCape on an application's GUI). Not widely supported yet. (-> Sven Drieling, several)
 - reworked all SVOject's file recognition
 - SVOjects:
 - JPEG: 68000 version did crash alyways, when loading or saving (-> Florian Zeiler)
 - FastILBM24: did crash with buffersizes larger 8K (-> several)
 - GIF, TIFF: misc (see there)
 - SVOperators:
 - AnyTo24Bit: HAM8 conversion still did not work correctly; forgot to fix (-> Florian Zeiler, Jürgen Schäfer)
 - Dither24Bit and more: see there
 - MISC:
 - some general fixes
 - Programmers:
 - updated documentation
 - bug-fixed and updated FD-Files
 - added Oberon2-Sources and -Includes by Sven Drieling
 - new error code(s)
 - Documentation:
 - fixed/updated IMPORTANT.doc

```
*****
* Revision history entries below V12.1 have been deleted *
*                               to save diskspace !      *
*****
```

1.25 AmiFIG.svobject

© 1996 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.1
Release Date : 07.06.1996

Description

~~~~~

AmiFIG.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support and a working GhostScript.svobject to be present.

It supports importing AmiFIG/Xfig vector drawings as bitmapped graphics. This is done by using fig2dev for conversion to Postscript (TM) files, which then will be parsed using a working Ghostscript installation with

GhostScript.svobject

.

AmiFIG.svobject will create a temporary file, which then will be parsed through superview.library again.

#### Reading :

(Any FIG drawings as long as the resulting Postscript (TM) commands are readable for Ghostscript).

#### Configuration

~~~~~

This FIG support module has been tested with:

- the AmiFIG 1.1 port from AmiNet: AmiFIG is (C) 1996 Andreas Schmidt AND NOT YET, because of missing fig2dev:
- the Xfig 1.9d amiga port from AmiNet, which has been done by Terje Pedersen (terjepe@login.eunet.no)

At first, you have to correctly install and configure Ghostscript and

GhostScript.svobject

.

After that you have to install AmiFIG/Xfig and either supply an AmigaDOS search path to a directory, where the executable "fig2dev" is stored (default), or explicitly specify path and/or name of the program by the controlpad entry AMIFIG_PATH (see below).

ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/AmiFIG.controlpad"  
ControlPad-Commands : - AMIFIG\_PATH=<fig2dev command path and/or name>  
; how fig2dev is to be called  
; e.g. AMIFIG\_PATH=Work:AmiFIG/fig2dev  
; default is: fig2dev

#### History

---



```
~~~~~
V3.1 (07.06.1996) :

- first version
```

## 1.26 ILBM.svobject

© 1994-96 by Andreas R. Kleinert.  
FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.6  
Release Date : 08.06.1996

### Description

```
~~~~~
ILBM.svobject is an external library module for the superview.library,
which needs any SVDriver with Bitplane-Support.
```

It supports reading and writing of IFF-ILBM Files.  
In detail these are :

#### Reading :

- IFF-ILBM uncompressed 1..8/24 Bit
- IFF-ILBM CmpByteRunl compressed 1..8/24 Bit

#### Writing :

- IFF-ILBM uncompressed 1..8/24 Bit
- IFF-ILBM CmpByteRunl compressed 1..8/24 Bit

Reading/writing from/to ClipBoard is supported.

### ControlPad-Switches

```
~~~~~
ControlPad-Name : "ENV:superview-library/ILBM.controlpad"
ControlPad-Commands : - ANIM_BODIES
 ; if set, ANIM files' ILBM-BODY chunks
 ; (first frame) will be extracted when
 ; such a file is encountered. Otherwise
 ; ANIM.datatype may do that task later.
```

### History

```
~~~~~
V3.6 (08.06.1996) :

- fixed bug in initialization code
- fixed chunk recognition (may have read beyond ID array size)
- fixed Clipboard file recognition (BMHD had to be first chunk)
- now supports reading of ANIM files' first frame (optional)

V3.5 (13.04.1996) :

- now finally, ultimately, absolutely fixed ILBM buffersize
```

calculation stuff (simply had been blind - thanks to  
-> Simon Edwards)

- when saving 24 Bit ILBMs, the buffer for 24 Bit oneplane to bitplane conversion was not correctly aligned, thus a little bit to small (mung wall hits) - the conversion buffer for reading on the other hand was a little bit to large, which did not actually matter. Fixed.  
(-> Simon Edwards)

#### V3.4 (14.03.1996) :

- IMPORTANT fix: argh! ILBM just had the same, nasty "not more than 256 bytes per line" bug as PCX had in former versions. It was so simple, that I didn't even saw it. Fixed now. Side-effect: buffer perhaps now always will be in available 32 Bit Fast Ram :-)  
Concerned reading and writing.
- no longer searches for 24 Bit viewmodes (ask for 8 Bit)
- completely rewrote decoding routines for unpacked (interleaved -> planebuffer) and packed (CmpByteRun1 -> planebuffer) routines.  
No longer using mixed UWORDS and ULONGs, only ULONGs and two UBYTES for counting, loops, etc.  
This may slow down processing on plain 68000 systems, but will a) increase speed on 680x0 and b) reduce possible casting errors and/or overflow errors caused by the compiler or utility library. Hope so.
- fixed progress indicator update intervals (reading and writing)
- writing: no longer depends on valid source (or temporary) svgfx\_BufferSize (would have caused mungwall hits when wrongly initialized). Also fixed buffersize calculation.

#### V3.3 (27.01.1996) :

- added REV. 2 of new, rock-solid startup code
- now will reject graphics with depths between 9..23 (e.g. weird 12 Bit formats) - which had not been supported anyway, but might have caused crashes
- using new GST

#### V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56
- fixed to correctly load masked images
- fixed size given in svgfx\_BufferSize

#### V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****  
* Revison history entries before V3.1 have been deleted for space reasons *  
*****
```

## 1.27 ACBM.svobject

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.4

Release Date : 08.06.1996

Description

~~~~~

ACBM.svobject is an external library module for the superview.library, which needs any SVDriver with Bitplane-Support.

It supports reading and writing of IFF-ACBM Files.

In detail these are :

Reading :

- IFF-ACBM uncompressed 1..8 Bit

Writing :

- IFF-ACBM uncompressed 1..8 Bit

Reading/writing from/to ClipBoard is supported.

History

~~~~~

V3.4 (08.06.1996) :

- fixed bug in initialization code
- fixed chunk recognition (may have read beyond ID array size)
- fixed Clipboard file recognition (BMHD had to be first chunk)

V3.3 (27.01.1996) :

- added REV. 2 of new, rock-solid startup code
- now will reject graphics with depths greater than 8 (e.g. 24 Bit graphics) - which had not been supported anyway, but might have caused crashes - when reading and writing
- using new GST

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

\*\*\*\*\*  
\* Revison history entries before V3.1 have been deleted for space reasons \*  
\*\*\*\*\*

## 1.28 Datatypes support

Loading Graphics via DataTypes under OS3

=====  
In General

-----  
The library generally also accesses OS3-Datatypes, if none of the installed SVOjects was able to identify a specific graphics file format.

Usually by default only DataTypes of class "picture" will be accepted.

This means, no Samples, Texts, AmigaGuide File or whatever else will be loaded and displayed via superview.library - by default.

But it is possible to enable these also - see configuration section for more information on how to change settings - so that for example introduction pictures of ANIMs may as well be loaded.

Specific Problems

-----  
As you may have expected, it is as well possible to replace single SVOjects by the corresponding DataTypes in case you think that the Datatype does work better or more reliable.

For temporarily enabling this, you would just have to "Remove" the SVOject and load the Datatype.

For permanently enabling this, you would have to delete the SVOject from "LIBS:svobjects".

But remember, that DTs in general cannot export 24 Bit Data and are not able to save any graphics (other than as IFF-ILBM).

Ralph Schmidt's picture.datatype V43 does allow loading of 24 Bit pictures by enhancing the datatype system with additional functionality. To be able using these features with SuperView, you need a registered CyberGraphX installation and CyberDataType.

On the other hand DO NOT TRY to run picture.datatype V43 without CyberDataType being installed. SuperViewLibrary's internal DataTypes reader likes accessing bitmaps directly - but CyberGraphX bitmaps may be somewhat stranger than the usual ones.

Precisely, SuperViewLibrary does touch DT bitmaps only, when trying to extract these as GfxBuffers.

## 1.29 PCX.svobject

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.4

Release Date : 14.3.1996

---

## Description

~~~~~

PCX.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support.

It supports reading and writing of PCX Files.

In detail these are :

Reading :

2-256 Colors, 24 Bit. In detail:

- PCX V2.5 upto 16 Colors (supports 4/8 color EGA/VGA palette)
- PCX V2.8a upto 16 Colors (supports 4/8 color EGA/VGA palette)
- PCX V2.8b upto 16 GreyScales
- PCX V3.00 upto 256 Colors
- PCX V2.5 - V3.00 with 24 Bit Data (see Remarks !)

Writing :

Depending on the Colordepth of the source the following is written :

Source Colors	Version	Type	Destination Colors
2 .. 16	PCX V2.8a	planar bitmap	16
32 .. 256	PCX V3.00	chunky pixel	256
(24 Bit)	PCX V3.00	RGB planar	(24 Bit)

For 2-256 Colors It is always tried, to write the files RLE-encoded, but if encoding is ineffective (output data nearly as large or even larger than input data), the files will be written unencoded.

24 Bit files will currently only be written unencoded.

History

~~~~~

V3.4 (14.3.1996) :

- fixed safety buffer calculation (as for ILBM, just with 64)

V3.3 (20.12.1995) :

- could not handle planar graphics (b&w or upto 16 Colors) with a width greater than 2048 (256\*8 -> buffer overflow). Fixed.
- (-> Jürgen Schäfer)
- added REV. 2 of new, rock-solid startup code
- no longer causes crashes
- using new GST

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

## 1.30 SVG.svobject

© 1994-96 by Andreas R. Kleinert.

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.4  
Release Date : 21.12.1995

Description  
~~~~~

SVG.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support.

It supports reading and writing of SVG Graphics Files, which is superview-Library's own graphics file format with XPK support.

In detail these are :

Reading :

- SVG ONEPLANE 8/24 Bit
BITPLANE 1..8 Bit

Writing :

- SVG ONEPLANE 8/24 Bit
BITPLANE 1..8 Bit

ControlPad-Switches
~~~~~

```
ControlPad-Name      : "ENV:superview-library/SVG.controlpad"
ControlPad-Commands : - PACKMETHOD=<xxxx>
                      ; if this one is specified, it is tried to
                      ; pack the resulting file with the specified
                      ; XPK-Packer. If this fails, the file keeps
                      ; unpacked.
```

File Format Information  
~~~~~

On "AmiNet:util/dtype" there's a separate package available, which consists of a "SVG.datatype" for applications not using superview.library, the description of the FileFormat and a conversion tool "AnyToSVG" (using Datatypes).

Some excerpts from there:

Pictures in the "SVG Graphics File Format" consist of two parts: a header and an attached xpk-packed or unpacked data file. Both parts are put into one single file.

Construction:

0x00	ID	UBYTE[18]	"SVG Graphics File"
0x12	Version	UWORD	always 1 yet
0x14	GfxDataOffset	ULONG	header length (depends on version)
0x18	LeftEdge	ULONG	
0x1c	TopEdge	ULONG	
0x20	Width	ULONG	
0x24	Height	ULONG	
0x28	ColorDepth	ULONG	actually available colors
0x3c	ViewMode32	ULONG	32 Bit ViewMode
0x40	PixelBits	UBYTE	1, 8, 24 (future: 16, 32)
0x41	PixelPlanes	UBYTE	# of planes with PixelBits
0x42	BytesPerLine	ULONG	bpl of a PixelPlane
0x46	ColorMap	UBYTE [256][3]	empty, if > 256 Colors

After that either follows XPK compressed data or uncompressed data, which can be detected by the leading chars "XPK" or "PP20" for packed data at GfxDataOffset (relative to beginning of the file).

History

~~~~~

V3.4 (21.12.1995) :

- added REV. 2 of new, rock-solid startup code

V3.3 (10.12.1995) :

- added new, rock-solid startup code
- no longer causes crashes
- changed some includes and so on
- now opens "xpkmaster.library" just when needed and not when initializing module. This should have two effects:
  - if you want to save packed data, but there's no library, it will be saved unpacked instead
  - initialization time should be much shorter, so that possibly the crashes occurring on 060 systems should no longer occur (hopefully)
  - the xpk-libraries may be flushed from memory completely, when no longer actually being needed and used
  - opening the sobject does no longer depend on the presence of xpk- and or powerpacker-library (so unpacked files may as well be read/written on systems without these)

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

V2.1 (24.09.1995) :

- replaced SVO.sobject by this one, which is smarter and works

a little bit more transparent for the user. Needs VMEM: assignment.

## 1.31 GIF.svobject

© 1994-96 by Andreas R. Kleinert.

(Also see notes under "Credits" and LZW notes under "Copyright".)

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.5  
Release Date : 15.06.1996

Description

~~~~~

GIF.svobject is an external library module for the superview.library, which needs any SVDriver with Oneplane-Support.

For using this module, LZW support must be globally enabled.

It supports reading and writing of GIF Files (87a and 89a derivatives are supported, where additional features of 89a will just be ignored).

In detail these are :

Reading :

- GIF 87a
- GIF 89a

Writing :

- GIF 87a
- GIF 89a

Source Colors	Version	Type	Destination Colors
2 .. 256	GIF 87/89a	chunky pixel	2 .. 256

Bugs and other Problems

~~~~~

Decoding is reentrant, encoding isn't:

Please note, that the Encoding Routines of the Library are not (yet) fully re-entrant, so that only one Task may Encode a picture at a time. The Library itself manages it, that the other Task has to Delay() until the Encoding Routines are "free" again.

This makes NOT REALLY A MATTER, because Encoding takes so much of the processor's time, that it is NOT EFFICIENT to ENCODE TWO OR MORE pictures AT THE SAME TIME ANYWAY !

ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/GIF.controlpad"
ControlPad-Commands : - GIF89A_TRANSPARENT_COLOR = <0..255>
; Default color to be enabled and set
; (graphics control extension block)

; Default: (not active)

History :

~~~~~

V3.5 (15.6.1996) :

- fixed bug, which may have caused upto 4 bytes of the library (base) to possibly be overwritten
- now allows writing of transparent GIFs (possible with GIF89a only); useful for creation of HTML pages. Transparent color is both, enabled and set, by a new controlpad entry.  
(-> Patrik Nydensten, Sven Drieling)

V3.4 (16.3.1996) :

- added LZW license check

V3.3 (21.12.1995) :

- now will correctly handle (reject) GIF files, which actually don't contain any graphics data  
(-> Jürgen Schäfer)  
There had been a sample GIF89a file, which only did contain comments and such stuff and caused an endless loop with deadlock.
- added REV. 2 of new, rock-solid startup code
- no longer causes crashes
- using new GST

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (09.10.1995) :

- upgraded to V3 specs
- correct error code with 24 Bit input  
(-> Florian Zeiler)
- and more

\*\*\*\*\*  
\* Revison history entries before V3.1 have been deleted for space reasons \*  
\*\*\*\*\*

## 1.32 BMP.svobject

© 1994-96 by Andreas R. Kleinert.

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.4

Release Date : 29.04.1996

Description

~~~~~

BMP.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support.

It supports reading and writing of Windows (TM) and OS/2 (TM) BitMap files (BMP). Not all derivatives might be supported, though. Extraction of Bitmaps from RIFF-BMP files (DIB) also is possible.

In detail these are :

Reading :

- unencoded BMP "wallpapers" in 1, 4, 8, 24 Bit ColorDepth.
- RLE-encoded BMP "wallpapers" in 4, 8 Bit ColorDepth.

Writing :

Depending on the Colordepth of the source the following is written :

Source Colors	Version	Type	Destination Colors
2	BMP (misc)	packed chk. pix.	2
4 .. 16	BMP (misc)	packed chk. pix.	16
32 .. 256	BMP (misc)	chunky pixel	256
(24 Bit)	BMP (misc)	RGB pixel	(24 Bit)

History

~~~~~

V3.4 (29.04.1996) :

- now can read RIFF-BMP files (DIB)

V3.3 (28.01.1996) :

- added REV. 2 of new, rock-solid startup code
- could not read BMPs with differing header and bitmap header size - now will deal correctly with the offsets (-> Grant Fribbens, Klaus Ziereis via Florian Zeiler)
- changed file format name to "BMP (Win,OS/2)"
- added read support for RLE4 compression
- using new GST

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

\*\*\*\*\*  
 \* Revison history entries before V3.1 have been deleted for space reasons \*  
 \*\*\*\*\*

### 1.33 Winlcon.svobject

© 1994-96 by Andreas R. Kleinert.  
 FREEDWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3  
 Release Date : 09.06.1996

#### Description

~~~~~

WinIcon.svobject is an external library module for the superview.library, which needs any SVDriver with Oneplane-Support.

It supports reading of W*nd*ws V3.x (TM) Icon Files.
 In detail these are :

Reading :

- WinIcon files with exactly 16 Colors
 (more than one Icon per file should work, but hasn't been tested yet.)

History

~~~~~

V3.3 (09.06.1996) :

- added new, rock-solid startup code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

V2.3 (22.07.1995) :

- reworked to close files immediately after reading/writing - if successful
- enabled immediate SVDriver switching

\*\*\*\*\*  
 \* Revison history entries before V2.2 have been deleted for space reasons \*  
 \*\*\*\*\*

## 1.34 FBM.svobject

© 1994-96 by Andreas R. Kleinert.  
 (Also see notes under "Credits".)  
 FREEDWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3  
 Release Date : 09.06.1996

#### Description

~~~~~

FBM.svobject is an external library module for the superview.library, which needs any SVDriver with Oneplane-Support.

It supports reading and writing of FBM (*nix) Files.
In detail these are :

Reading :

- FBM 8 Bit x 1 Plane: 2..256 Colors/256 Greyscales
- FBM 8 Bit x 3 Planes: 24 Bit

Writing :

- FBM 8 Bit x 1 Plane: 2..256 Colors/256 Greyscales
- FBM 8 Bit x 3 Planes: 24 Bit

Following to the specifications, an 2..128 Colors file takes as much space as an 256 Colors file : always 8 Bits are written.
Only the ColorMap is sized differently for different ColorDepths.

History

~~~~~

V3.3 (09.06.1996) :

- added new, rock-solid startup code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

V2.11 (29.07.1995) :

- reworked to close files immediately after reading/writing - if successful
- enabled immediate SVDriver switching
- fixed generation of ViewModes (important for 24 Bit and 8 Bit with ECS)

\*\*\*\*\*  
\* Revison history entries before V2.11 have been deleted for space reasons \*  
\*\*\*\*\*

## 1.35 PNM.svobject

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3  
Release Date : 09.06.1996

Description

~~~~~

PNM.svobject is an external library module for the superview.library,
which needs any SVDriver with Oneplane-Support.

It supports reading and writing of PNM (*nix) Files.
In detail these are :

Reading :

All binary (non-ASCII) derivatives of the PNM format will be read (P4/P5/P6), non-binary (ASCII) derivatives are not supported (P1/P2/P3).

- PBM (P4) Black & White
- PGM (P4) 256 GreyScales
- PPM (P6) 24 Bit TrueColor

Writing :

All input data, no matter how many Colors actually are used, will be written as binary 24 Bit PPM data (P6) - this is the only way to prevent color information from getting lost.

History

~~~~~

V3.3 (09.06.1996) :

- added new, rock-solid startup code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

V2.5 (02.08.1995) :

- some P? files written by foreign software have not been recognized due to some " " replaced by 0x0a in the header (which is not a problem when using scanf() for getting the values). We now do check for all the three: " ", 0x0a and 0x0d.

\*\*\*\*\*  
\* Revison history entries before V2.4 have been deleted for space reasons \*  
\*\*\*\*\*

## 1.36 PNG.svobject

© 1996 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2

Release Date : 15.06.1996

Description

~~~~~

PNG.svobject is an external library module for the superview.library,

which needs any SVDriver with Oneplane-Support.

It supports reading of Portable Network Graphics (PNG - pronounce it "PiNG", if you like ;-) files.

In detail these are :

Reading :

- 8 Bit color mapped files
- True color files (16, 24, 32 Bit) as 24 Bit (perhaps 48 Bit as well)

History :

~~~~~

V3.2 (15.06.1996) :

- fixed file format description string

V3.1 (10.02.1996) :

- first version

## 1.37 C64.svobject

© 1994-96 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3

Release Date : 09.06.1996

Description

~~~~~

C64.svobject is an external library module for the superview.library, which needs any SVDriver with Oneplane-Support.

It supports reading of C64 Graphics Files (Koala, Doodle).

In detail these are :

Reading :

Format	Dimensions	Colors
Doodle	320x200	2/16
Koala	160x200 -> 320x200	4/16

History

~~~~~

V3.3 (09.06.1996) :

- added new, rock-solid startup code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revision history entries before V3.1 have been deleted for space reasons *
*****
```

## 1.38 CDR.svobject

© 1996 by Andreas R. Kleinert.

FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.1  
Release Date : 07.06.1996

### Description

~~~~~

CDR.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support and BMP.svobject to be present.

It supports reading of the previews which may be part of CorelDraw .CDR files (e.g. with version 4) - these are related with plain Windows BMP files, so that CDR.svobject will extract the graphics data, create a temporary file, write it as a BMP file and then parse it through superview.library again.

(See documentation of BMP.svobject for which BMP derivatives currently are supported).

Reading :

(Any contained preview, as long as the specific contained BMP derivative is supported by superview.library).

History

~~~~~

V3.1 (07.06.1996) :

- first version

## 1.39 IMG.svobject

© 1994-96 by Andreas R. Kleinert.

FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3  
Release Date : 09.06.1996

### Description

~~~~~

IMG.svobject is an external library module for the superview.library, which needs any SVDriver with Bitplane-Support.

It supports reading of IMG (GEM/Metafile) Files.
In detail these are :

Reading :

- IMG (16 byte header) greyscaled (1..8 bits = 2..256 scales)
- IMG (18 byte header) greyscaled (1..8 bits = 2..256 scales)

History

~~~~~

V3.3 (09.06.1996) :

- added new, rock-solid startup code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

V2.9 (29.07.1995) :

- reworked to close files immediately after reading/writing - if successful
- enabled immediate SVDriver switching
- fixed generation of ViewModes (important for 8 Bit with ECS)

\*\*\*\*\*  
\* Revison history entries before V2.9 have been deleted for space reasons \*  
\*\*\*\*\*

## 1.40 TIFF.svobject

© 1994-96 by Andreas R. Kleinert.

(Also see notes under "Credits" and LZW notes under "Copyright".)

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.6  
Release Date : 15.03.1996

Description

~~~~~

TIFF.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support.

It supports reading and writing of TIFF-Files (V5.0).
In detail these are :

Reading :

Compression	Code	PixelBits	Planes	max. Depth	Storage Type
None	1	1	(system)	1*(system)	BITPLANE
None	1	8	1,3	8 / 24	ONEPLANE
Mac PackBits	32773	1	(system)	1*(system)	BITPLANE
Mac PackBits	32773	4	1	4	ONEPLANE
Mac PackBits	32773	8	1	8	ONEPLANE
LZW	5	8	1,3	8 / 24	ONEPLANE

Not all of these modes may always work correctly. Please report any bugs. For supporting LZW compression, LZW support must be globally enabled for license reasons.

Writing :

```

Depth                written as

upto 256 Colors      Motorola (MM), None, 8, 1, ONEPLANE
(24 Bit)             Motorola (MM), None, 8, 3, ONEPLANE

```

Remarks

~~~~~

- Files with unknown compression types (e.g. CCITT Fax) will as well be rejected as files with unsupported Compression/PixelBits/Plane combinations, like e.g. None/4/2.
- If you have some FREELY DISTRIBUTABLE TIFF-Files, which do not work with this program, you may send them to me, so that I can try to improve this module to also allow reading of these.

History

~~~~~

V3.6 (16.03.1996) :

- added LZW license check

V3.5 (13.03.1996) :

- Intel style 24 Bit RGB graphics, which had been placed as BGR in an RGB buffer (red and blue swapped). Fixed for this special case. (-> Lothar Köster via Florian Zeiler)

V3.4 (10.02.1996) :

- added REV. 2 of new, rock-solid startup code
- using new GST
- would not have tried to actually decode LZW TIFFs with only one single strip of data, which resulted in a black screen - now it tries, but maybe will still not be able to unpack these (strangely compressed LZW EPS-Headers of FW).

V3.3 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.2 (12.10.1995) :

- now also able to read multi-strip PackBits files
(-> Mr. Reichert via Jürgen Schäfer)
- changed order of writing TIFF Tags according to specs (ascending)

V3.1 (10.10.1995) :

- upgraded to V3 specs
- changed number of tags for written TIFF files; more information:
 - 8 Bit: now includes photometric tag
 - 24 Bit: now includes photometric and planar config tag
 Should enable even W*RD and C*r*lDr*w to read them
(-> Florian Zeiler)
- also did write bad (incomplete) header, which e.g. caused programs to think it was a multiple picture file (missing 0L at end of image directory)
- reader module now calculates buffersize better

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

1.41 EPS.svobject

© 1994-96 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.4
Release Date : 07.06.1996

Description

~~~~~

EPS.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support and TIFF.svobject (respectively GhostScript.svobject) to be present.

It allows two operations, depending on controlpad preferences settings:

- extraction of the trailers/previews which may be part of EPS(F) files. This usually will be TIFF graphics, but it makes no matter, which file format actually is included, because EPS.svobject will create a temporary file, which then will be parsed through superview.library again.  
(See documentation of TIFF.svobject for which TIFF derivates currently are supported).
- extraction of the Postscript (TM) part of EPS(F) files, which then, in form of a temporary file, will be parsed through superview.library (and perhaps also GhostScript.svobject) again.

Reading :

- Any contained trailers/previews/thumbnails, as long as the specific contained file format - if any - is supported by superview.library

- Postscript (TM) as far as supported by the installed Ghostscript implementation

#### Writing:

- EPS files with Postscript (TM) image content, without TIFF header (only from 24 Bit sources -> use the AnyTo24Bit operator).

The graphics will NOT be vectorized - instead a bitmap-like representation of it will be translated into Postscript (TM) format.

#### Remarks

~~~~~

Please note, that the Encoding Routines of the Library are not (yet) fully re-entrant, so that only one Task may Encode a picture at a time. The Library itself manages it, that the other Task has to Delay() until the Encoding Routines are "free" again.

ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/EPS.controlpad"
ControlPad-Commands : - EXTRACTMODE=<HEADER|PS>
                      ; Determines, whether the preview image or
                      ; the Postscript (TM) part should be extracted,
                      ; while parsing the Postscript (TM) part requires
                      ; a working Ghostscript installation with
                      ; GhostScript.svobject being correctly configured
                      ; default is: HEADER
- SAVE_ROTATE=<ON|OFF>
  ; by default, the picture seems to be rotated
  ; with Postscript (TM) output of this module.
  ; Enabling SAVE_ROTATE will again re-rotate it to
  ; the original position.
  ; default is: OFF
- SAVE_CENTER=<ON|OFF>
  ; centers the images on the page with the
  ; desired size of SAVE_WIDTH x SAVE_HEIGHT (or not)
  ; default is: ON
- SAVE_RLE=<ON|OFF>
  ; allows to apply RLE compression to the
  ; PS output
  ; default is: OFF
- SAVE_DPI=<value>
  ; DPI value to use for the output
  ; default is: 300
- SAVE_WIDTH=<value>
  ; Width of the PS page
  ; default is: 612
- SAVE_HEIGHT=<value>
  ; Width of the PS page
  ; default is: 762
```

#### History

~~~~~

V3.4 (07.06.1996) :

- added new, rock-solid startup code
- fixed bug in initialization code
- now optionally allows extraction of Postscript (TM) part of EPS files, which requires Ghostscript and GhostScript being correctly installed and configured
- now is able to WRITE EPS files from 24 Bit sources: no trailer, but with Postscript (TM) content

V3.3 (27.04.1996) :

- er...well. Know, that something was changed, but don't know what.

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

1.42 GhostScript.svobject

© 1996 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.1

Release Date : 07.06.1996

Description

~~~~~

GhostScript.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support and PNG.svobject to be present.

It supports importing Postscript (TM) language and PDF files as bitmapped graphics. This is done by using Ghostscript for conversion, giving a DPI resolution and colordepth for the only supported output file format PNG.

GhostScript.svobject will create a temporary file, which then will be parsed through superview.library again.

(See documentation of PNG.svobject for which PNG derivates currently are supported).

Reading :

(Any Postscript (TM) and PDF files as long as readable for Ghostscript).

## Configuration

~~~~~

This Ghostscript support module has been tested with the Ghostscript 3.53 port from AmiNet, with amiga driver & no ixemul.library: Aladdin Ghostscript is Copyright (C) 1989, 1995 Aladdin Enterprises. All rights reserved.

At first, Ghostscript must have successfully and working been installed (test it via command line access).

For enabling Ghostscript support, you either have to supply an AmigaDOS search path to a directory, where the executable "gs000" is stored (default), or explicitly specify path and/or name of the program by the controlpad entry GS_PATH (see below).

Setting-up Ghostscript

~~~~~

Since the documentation for Ghostscript 3.53 is anything-but-not-very-verbose I decided, to shortly explain here, what steps to perform to successfully install Ghostscript on your system:

- unpack Ghostscript to where you wish it to be located at (e.g. "Work:Ghostscript/")
- add an assign called "Ghostscript:" to your S:User-Startup, which points to that directory (e.g. "Assign Ghostscript: Work:Ghostscript")
- create a directory Ghoscript:fonts (e.g. type "Makedir Ghostscript:fonts" in a shell)
- create four environment variables at both locations, ENV: and ENVARC:

| Variable    | Content           |
|-------------|-------------------|
| GS_DEVICE   | amiga             |
| GS_FONTPATH | Ghostscript:fonts |
| GS_LIB      | Ghostscript:data  |
| GS_OPTIONS  |                   |

Respect the case, and please note, that the GS\_OPTIONS variable actually is empty, but should be present.

You should use "SetEnv ENV:var content" and "SetEnv ENVARC:var content" commands for creating these variables.

Please note, that there are many different ways of installing Ghostscript (an assign is not actually needed, and no one actually must supply a fonts directory), but this way it definitely should \*work\*

## Remarks

~~~~~

- please note, that setting DPI to higher values does not necessarily increase image quality, but memory usage (default is DPI=72). Same is to mention for the output color depth: 256 colors give better results, when you've to perform dithering.
- files to be recognized must either have the standard Postscript (TM)

- header (containing '%!PS-Adobe' in it) or have the file extension ".ps" and begin with '%' (containing instructions for GhostScript).
- PDF files are also supported (extension ".pdf" and beginning with '%'), but having the correct and needed fonts installed is more critical for these (GS_FONTPATH variable of Ghostscript). Had not enough fonts to successfully test loading of any of these.
 - error output of Ghostscript still is directed to `stdio/Output()`, so problems may be visible transparently

ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/GhostScript.controlpad"
ControlPad-Commands : - GS_PATH=<gs command path and/or name>
                       ; how Ghostscript is to be called
                       ; e.g. GS_PATH=Ghostscript:gs000
                       ; default is: gs000
- OUTPUTMODE=<PNG256|PNG24BIT>
  ; Output file format to be used
  ; (-sDEVICE parameter of Ghostscript's gs).
  ; Only PNG supported yet.
  ; default is: PNG256
- DPI=<value>
  ; DPI value to use for the converted output
  ; (-r parameter of Ghostscript's gs)
  ; default is: 72
```

#### History

~~~~~

V3.1 (07.06.1996) :

- first version

1.43 Targa.svobject

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with `superview.library`.

```
Version      : 3.3
Release Date : 07.06.1996
```

Description

~~~~~

Targa.svobject is an external library module for the `superview.library`, which needs any `SVDriver` with `Bit-/Oneplane-Support`.

It supports reading and writing of TGA (Truevision Targa) Files.

In detail these are :

Reading :

| Colors | Depth | Organisation                | RLE-Compression |
|--------|-------|-----------------------------|-----------------|
| 1      | 1     | planar monochrome           | supported       |
| 256    | 8     | chunky pixel (colors/gray)  | supported       |
| 32768  | 16    | "HighColor 15/16 Bit" Pixel | supported       |

(24 Bit)      BGR-Pixel      supported

Writing :

Depending on the Colordepth of the source the following is written :

| Source Colors | Type                      | Destination Colors |
|---------------|---------------------------|--------------------|
| 2 .. 256      | Uncompressed chunky pixel | 256                |
| (24 Bit)      | Uncompressed BGR          | (24 Bit)           |

Remarks

~~~~~

- 32 Bit graphics files are not supported yet.
 - ColorMaps have to be of type "3-Byte BGR".
 - The flags for "mirroring" Images vertically and/or horizontally are not fully interpreted yet, nevertheless these will be reported by SVL_FileInfoRequest().
- If the VERTINV flag is not set, the picture will be assumed to be written as "from bottom to top", otherwise as "from top to bottom". Some programs do not set these flags right, when writing, so that you might get just the opposite result as expected.
- The HORIZINV flag is currently ignored: when reading such a picture as usual, you'd get a mirrored image. But this flag is also set wrong sometimes ...

History

~~~~~

V3.3 (07.06.1996) :

- added new, rock-solid startup code
- fixed bug in initialization code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

\*\*\*\*\*  
 \* Revison history entries before V3.1 have been deleted for space reasons \*  
 \*\*\*\*\*

## 1.44 WPG.svobject

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version        : 3.3  
 Release Date : 09.06.1996

Description

~~~~~

WPG.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support.

It supports reading of WPG (WordPerfect) BitMap Files. In detail these are :

Reading :

- WPG Bitmap (greyscaled) with 1, 2, 4 or 8 bits colordepth
- WPG Bitmap (colorMap) with 1, 2, 4 or 8 bits colordepth

Remarks

~~~~~

WordPerfect WPG Files do not necessarily have to contain bitMap graphics, they also may contain various other data, e.g. vector graphics. If a WPG file contains a bitMap graphic in any of its chunks, it will be loaded, otherwise the file will be rejected.

If a file does not contain any color information, WPG.svobject will generate greyscales by default. This will be mentioned in the file-info requester.

This version actually has been tested with graphics with 1, 4 and 8 Bit colordepth (2, 16 and c56 Colors). Due to the fact, that the 2 bit-routine is identically to the 1 bit-routine you should not get any problems with those pictures.

History

~~~~~

V3.3 (09.06.1996) :

- added new, rock-solid startup code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

1.45 SunRaster.svobject

© 1994-96 by Andreas R. Kleinert.

FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3
Release Date : 07.06.1996

Description

~~~~~



SunRaster.svobject is an external library module for the superview.library, which needs any SVDriver with Bit-/Oneplane-Support.

It supports reading and writing of SunRaster (RAS) Files.  
In detail these are :

Reading :

- SunRaster with 2 Colors ( 1 Bit, planar)
- SunRaster with 256 Colors ( 8 Bit, chunky pixel)
- SunRaster with 24 Bit (24 Bit, R-G-B)

Writing :

Depending on the Colordepth of the source the following is written :

| Source Colors      | Type                                                 | Destination Colors |
|--------------------|------------------------------------------------------|--------------------|
| 2..256<br>(24 Bit) | Uncompressed Chunky Pixel<br>Uncompressed 24 Bit RGB | 256<br>(24 Bit)    |

Remarks

~~~~~

- files with Colorbits other than 1, 8 or 24 are not supported yet
- only RGB-planar colormaps supported (or monochrome, without map)
- max. 1 plane of bitmap data allowed

History

~~~~~

V3.3 (07.06.1996) :

- added new, rock-solid startup code
- fixed bug in initialization code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

## 1.46 SGI.svobject

© 1994-96 by Andreas R. Kleinert.

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3  
Release Date : 07.06.1996

Description

~~~~~

SGI.svobject is an external library module for the superview.library, which needs any SVDriver with 24 Bit Oneplane-Support.

It supports reading and writing of SGI Files.
In detail these are :

Colors Depth	Organisation	RLE-Compression
256 8	8 Bit Chunky Pixel (gray)	supported
(24 Bit)	8:8:8 24 Bit RGB	supported

Writing :

Source Colors	Type	Destination Colors
(24 Bit)	Uncompressed RGB	(24 Bit)

Remarks

~~~~~

- Currently files with BPC-Values of 2 are not supported, which means that pixel data has to be stored in bytes, not words (either gray 8 Bit or 24 Bit RGB as 8:8:8, not 16 or 16:16:16)

History

~~~~~

V3.3 (07.06.1996) :

- added new, rock-solid startup code
- fixed bug in initialization code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

* Revison history entries before V3.1 have been deleted for space reasons *

1.47 PICT.svobject

© 1995-96 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.5
Release Date : 21.05.1996

Description

~~~~~

PICT.svobject is an external library module for the superview.library,

which needs any SVDriver with 24 Bit Oneplane-Support.

It supports reading of Mac PICT-2 Metafile graphics.  
In detail these are :

Reading :

- Mac PICT-2 with 1, 2, 4, 8 or 24 Bit ColorDepth  
always as 24 Bit RGB Data (JPEG also as dithered 8 Bit)

Remarks

~~~~~

- the file extension has to be ".pct" (as on PCs), ".pic" or ".pict", otherwise it will be rejected (there do more checks take place, but those are less reliable)
- pictures always will be exported as 24 Bit RGB data
- font handling always will operate with the internal standard font instead of the appropriate ones (derived from PBM package, see Credits)

Limitations

~~~~~

Please note, that the Decoding Routines of the Library are not (yet) fully re-entrant, so that only one Task may Decode a picture at a time. The Library itself manages it, that the other Task has to Delay() until the Decoding Routines are "free" again. Since Decoding usually is managed very fast, this should not actually matter.

History

~~~~~

V3.5 (21.05.1996) :

- fixed bug, which may have caused upto 8 bytes of the library (base) to possibly be overwritten

V3.4 (21.12.1995) :

- added REV. 2 of new, rock-solid startup code

V3.3 (10.12.1995) :

- added new, rock-solid startup code
- replaced IEEE with FFP support (due to crashes)
- no longer causes crashes
- now using GST and math pragmas (does it help anything ?)
- using common SAVES for the decoding routines

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56
- year of creation date wasn't '54 8-

V3.1 (07.10.1995) :

- upgraded to V3 specs
-

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

1.48 Pictor.svobject

© 1994-96 by Andreas R. Kleinert.
FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3
Release Date : 09.06.1996

Description

~~~~~

Pictor.svobject is an external library module for the superview.library,  
which needs any SVDriver with Bit-/Oneplane-Support.

It supports reading of Pictor/PC Paint (PIC) Files.  
In detail these are :

### Reading :

- Files with 1, 4, or 8 Bit ColorDepth  
(monochrome or with EGA or VGA palette).

### History

~~~~~

V3.3 (09.06.1996) :

- added new, rock-solid startup code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

1.49 MAC.svobject

© 1994-96 by Andreas R. Kleinert.
FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3
Release Date : 09.06.1996

Description

~~~~~

MAC.svobject is an external library module for the superview.library, which needs any SVDriver with Bitplane-Support.

It supports reading of MAC (MacIntosh MacPaint) Files.  
In detail these are :

Reading :

- MAC Black & White 576x720

Remarks

~~~~~

MacPaint Files, which are to be loaded into "MAC.svobject" should contain the specific MacBinary Header (first the 128 Byte-Header, then the MacPaint specific data appended to it).

In the PC area, there may sometimes files be found, which just contain the MacPaint 576x720 black and white Data.

These files - without the header - can only be identified via the three leading zero bytes at the beginning of the 512 Byte MacPaint header.

But there's never a 100% guarantee that a File with three leading zeroes really is a MacPaint File, so we also request a filename ending with ".mac" or ".MAC" in this special case.

History

~~~~~

V3.3 (09.06.1996) :

- added new, rock-solid startup code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

\*\*\*\*\*  
\* Revison history entries before V3.1 have been deleted for space reasons \*  
\*\*\*\*\*

## 1.50 JPEG.svobject

© 1994-96 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREEMWARE. All rights reserved. Only to be distributed with superview.library.

This software is based in part on the work of the Independent JPEG Group.

Version : 3.7

Release Date : 10.03.1996

Description

~~~~~

JPEG.svobject is an external library module for the superview.library, which needs any SVDriver with Oneplane-Support.

It supports reading and writing of JPEG Files as written by "The Independent JPEG Group's JPEG Software" (release 6).

In detail these are :

Reading :

- JPEG (IJG-JFIF), with output in 256 Colors or 24 Bit

Writing :

- JPEG (IJG-JFIF), from upto 8 or 24 Bit Input

Remarks

~~~~~

\* VMEM:

\* Temporary data might be written to a directory assigned to "VMEM:", if necessary, but please note, that this might affect your HardDisk, if this program crashes (e.g. on a corrupt JPEG-picture) !  
So don't blame me, if this happens, but select a \_safe\_ place for this temporary-file directory !

\* Writing JPEG-Files :

\* Note, that JPEG-compression is lossy, which means that the original picture cannot be reconstructed totally.  
Only JPEG-compress files, of which you have backups, or files which you never want to edit and enhance again.  
You should always control the results from saving an JPEG-File : in some special cases (e.g. many thin lines on the screen) there may occur strong differences to the original picture.

ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/JPEG.controlpad"

ControlPad-Commands : - COLORDEPTH=<8|24>
; For READING colored 24 Bit images:
; should the output be done as 8 or 24 Bit Data ?
; default is 8 Bit data. Grayscaled graphics
; will _always_ be exported as 8 Bit
- FORCE_GRAY_DECODE
; For READING images:
; this switch overrides the selected depth
; for colored images and always exports them
; as 8 Bit grayscaled pictures
; default is: not set
- DECODE_METHOD=<FAST_INTEGER|SLOW_INTEGER|
FLOATING_POINT|FASTEST_INTEGER>
; For READING images:
; how to DECODE the 24 Bit or gray data
; default is: FAST_INTEGER
- DITHERMODE=<DITHER_FLOYD-STEINBERG|
NO_DITHERING|DITHER_ORDERED>
; For READING and dithering 24 Bit images
; to 8 Bit images:

```

; how and whether to DITHER the 24 Bit data
; default is: DITHER_FLOYD-STEINBERG
- COLOR_QUANTIZING=<FAST|SLOW>
; For READING and dithering 24 Bit images
; to 8 Bit images:
; whether to QUANTIZE the 24 Bit data within
; two passes
; default is: SLOW
- UPSAMPLING=<ON|OFF>
; For READING and dithering 24 Bit images
; to 8 Bit images:
; whether to do fancy upsampling on the 24 Bit
; default is: ON
- QUANTIZATION=<0..100>
; For WRITING 8/24 Bit images as 24 Bit JPEG:
; 0 - poor quality, big compression
; 100 - best quality, no compression
; never use such extreme values, but try
; something like 20, 50, 75
; default is: 75)
- FORCE_GRAY_ENCODE
; For WRITING images:
; this switch causes grayscales to be written
; no matter, whether the input was colored
; default is: not set
- ENCODE_METHOD=<FAST_INTEGER|SLOW_INTEGER|
                FLOATING_POINT|FASTEST_INTEGER>
; For WRITING images:
; how to ENCODE the 24 Bit or gray data
; default is: FAST_INTEGER
- OPTIMIZE=<ON|OFF>
; For WRITING 24 Bit images:
; whether to optimize the generated huffman
; code (good compression, but slow)
; default is: OFF
- PROGRESSIVE=<ON|OFF>
; For WRITING 24 Bit images:
; whether to write progressive JPEG files
; default is: OFF

```

History

~~~~~

V3.7 (10.03.1996) :

- changed jconfig.h: - explicite "VMEM:" reference
  - ALIGN\_TYPE now is "long"
- removed makefile references to mieee (irrelevant for svobject)
- added some \_\_saveds to hook functions (read + write)
- JPEGs with cinfo->image\_height != cinfo->output\_height will no longer cause problems (did these in the past ?)
- on an error in the JPEG code, our hook function re-jmp point now will also do a "finish\_decompress" if necessary and no longer cause an "internal", but a "decode error" (same with "compress" when writing)
- removed that nasty "signal catcher" stuff (useless and dangerous with shared library)
- error\_exit no longer calls output\_message, since we just did

- a "return;" there 8-)
- (read + write; write code even did not set output\_message at all :(
- cinfo and jerr not checked for NULL after allocation (read). Oops.  
(consequently, also \_never\_ delocated these)
- on writing errors, outfile may have been closed to soon
- if there wasn't enough memory for GfxBuffer allocation, decompression process would have been "finished", although it never "started"
- added safety value to JPEG decompression buffer  
(not included in svgfx\_BufferSize)
- removed "dead" buffer assignment (so the optimizer may do less worse ;-)

## V3.6 (10.02.1996) :

- had been linked with wrong math stubs.  
Fixed and recompiled completely.

## V3.5 (21.12.1995) :

- did again reference JPEGTMP:  
(-> Sven Drieling)
- added REV. 2 of new, rock-solid startup code

## V3.4 (10.12.1995) :

- added new, rock-solid startup code
- replaced IEEE with FFP support (due to crashes)
- no longer causes crashes
- MAJOR rewrote (finally succeeded)
- completely rewritten by using the IJG sources' release V6
- FASTER
- now fully reentrant for parallel uses within the  
Amiga's multitasking system
- MANY new options for specifying more detailed, which speed,  
quality or way of calculation should be used (controlpads)
- using common SAVEDS for the en-/decoding routines

## V3.3 (24.11.1995) :

- small changes
- no longer accepts HAM (use AnyTo24Bit, please)

## V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56
- no longer uses utility.library

## V3.1 (6.10.1995) :

- upgraded to V3 specs
- 68000 version did crash always when loading/saving  
(-> Florian Zeiler)

```
*****  
* Revison history entries before V3.1 have been deleted for space reasons *  
*****
```



## 1.51 PCD.svobject

© 1994-96 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3

Release Date : 07.06.1996

Description

~~~~~

PCD.svobject is an external library module for the superview.library, which needs any SVDriver with 24 Bit Oneplane-Support.

It supports reading of the unpacked resolutions of PCD-Files, as BASE/16 (192x128), BASE/4 (384x256) and BASE (768x512).

It DOES NOT READ the BASE*4 (1536x1024) and BASE*16 (3072x2048) resolutions, since these are encrypted in a way, which is not only undocumented, but also copyrighted by Kodak.

Legal Issues

~~~~~

This program module is FREEWARE.

The author does not claim any copyrights on the code, which is used to decode the YUV-data fromout the PCD-file (see credits) or on the other PCD-related information.

Copyright is only claimed for the program as a whole, which means that some parts of the library module, which are also used within other SVObjects, are copyrighted by the author.

If the writing or the publication of this program should ever be considered to be partly any kind of a violation of third party copyrights, it is hereby expressed that the usage of this program is only allowed, if any user of it agrees to the following:  
If the case as described above takes place, any user has to delete any copies of this program immediately when he gets informed about it.

ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/PCD.controlpad"

ControlPad-Commands : - OUTPUTFORMAT=<BASE/16 | BASE/4 | BASE >
; specifies the output resolution to be used

History

~~~~~

V3.3 (07.06.1996) :

- added new, rock-solid startup code
- fixed bug in initialization code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56
-

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

## 1.52 FastILBM24.svobject

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3  
Release Date : 07.06.1996

Description  
~~~~~

FastILBM24.svobject is an external library module for the superview.library, which needs any SVDriver with Oneplane-Support.

It has especially been written to allow fast and "low-memory" loading and dithering of ILBM 24 Bit graphics.

This means, that no compressed or uncompressed 24 Bit data will be completely loaded into memory, but directly be dithered to HAM6/8, so that even people with less memory and no graphics card will be able to view those nice 24 Bit graphics.

May be deactivated, even if installed; may be switched between HAM6 and HAM8; allows increasing/decreasing of internal cache buffer.

ControlPad-Switches
~~~~~

```
ControlPad-Name      : "ENV:superview-library/FastILBM24.controlpad"
ControlPad-Commands : - BUFFERSIZE=<Size in Bytes>
                       ; how many bytes should be buffered
                       ; (more = faster)
                       ; Default is 8192, minimum is 256
- PASSTHROUGH
  ; do not recognize ANY pictures, so that
  ; they e.g. might be passed through to
  ; ILBM.svobject instead
- DITHERMODE=<HAM6_QUICK|HAM8_QUICK>
  ; Default is HAM6_QUICK
```

History  
~~~~~

V3.3 (07.06.1996) :

- added new, rock-solid startup code
- fixed bug in initialization code

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs
 - BUFFERSIZE always was 8K, no matter what specified (caused crashes)

 * Revison history entries before V3.1 have been deleted for space reasons *

1.53 YUVN.svobject

© 1994-96 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.4
 Release Date : 08.06.1996

Description

~~~~~

YUVN.svobject is an external library module for the superview.library, which needs any SVDriver with 24 Bit Oneplane-Support.

It supports reading and writing of IFF-YUVN (YUVN) Files.  
 In detail these are :

Reading :

- IFF-YUVN 24 Bit (gray, 411, 422, 444, 211, 222)

Gray will be imported as 24 Bit with R=G=B.  
 Use "ExtractGrayScales" operator to reduce  
 memory usage - and redundancy

Writing :

- IFF-YUVN 24 Bit (411)

Reading/writing from/to ClipBoard is supported.

Remarks

~~~~~

- only 24 Bit sources will be written as IFF-YUVN.
 256 color graphics are NOT automatically transformed to 24 Bit,
 so that you may have to use the "AnyTo24Bit" operator before saving.

History

~~~~~

V3.4 (08.06.1996) :

- fixed bug in initialization code  
 - fixed chunk recognition (may have read beyond ID array size)  
 - changed Clipboard file recognition

V3.3 (28.01.1996) :

- now is able to distinguish between gray (400 with Y only and no U,V) and 411, 422, 444, 211, 222 when reading.  
Reading of 211 and 411 has been tested - the rest should work also (if not, take a look into the info requester and send me the graphics via snail mail).
- fixed reading of 411 (and added more types, see above)
- fixed writing of 411:
  - was wrong format for 411
  - YCHD chunk was too small (only 20 Bytes)
  - odd-width graphics had not been aligned (distorted display).  
Now will be aligned to four-byte boundary, but this naturally causes the (upto) three last bytes of a line to get an accidental color (not really: we set Y to 0 for all the pixel, but these are still dependent from U and V of the pixel at their left border, so that there will appear a "HAM effect").  
There does not exist a better way.
  - now reporting that "24 Bit input is needed" when trying to save 8 Bit bitplane data
  - changed file format name to "YUVN (411)"  
(-> Hans-Joachim Grund via Florian Zeiler)
- added REV. 2 of new, rock-solid startup code
- no longer causes crashes
- using new GST

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

## 1.54 DEEP.svobject

© 1995-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3  
Release Date : 08.06.1996

Description

~~~~~

DEEP.svobject is an external library module for the superview.library, which needs any SVDriver with 24 Bit Oneplane-Support.

It supports reading and writing of specific IFF-DEEP (DEEP) Files.
In detail these are :

Reading :

- IFF-DEEP 24 Bit RGB 8:8:8, uncompressed
- IFF-DEEP 24 Bit RGB 8:8:8, RLE-compressed

Writing :

- IFF-DEEP 24 Bit RGB 8:8:8, uncompressed

Reading/writing from/to ClipBoard is supported.

Remarks

~~~~~

- only 24 Bit sources will be written as IFF-DEEP.  
There's no conversion done from e.g. 256 Colors to 24 Bit to perform the requirements of this file format.
- other compression methods (e.g. Huffman) aren't yet supported due to a lack of information on implementation

History

~~~~~

V3.3 (08.06.1996) :

- added new, rock-solid startup code
- fixed bug in initialization code
- fixed chunk recognition (may have read beyond ID array size)
- fixed Clipboard file recognition (DGBL had to be first chunk)

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

1.55 QRT.svobject

© 1995-96 by Andreas R. Kleinert.

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.5
Release Date : 07.06.1996

Description

~~~~~

QRT.svobject is an external library module for the superview.library, which needs any SVDriver with 24 Bit Oneplane-Support.

It supports reading and writing of QRT (POV RayTracer) Files.

In detail these are :

Reading :

- QRT Dump 24 Bit

Writing :

- QRT Dump 24 Bit

Remarks

~~~~~

- only 24 Bit sources will be written as QRT.
There's no conversion done from e.g. 256 Colors to 24 Bit
to perform the requirements of this file format.

History

~~~~~

V3.5 (07.06.1996) :

- fixed bug in initialization code

V3.4 (21.05.1996) :

- changed error code for trying to save < 24 Bit

V3.3 (28.01.1996) :

- added REV. 2 of new, rock-solid startup code
- changed file format name to "QRT/POV Raytracer"
- using new GST

V3.2 (24.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (07.10.1995) :

- upgraded to V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

## 1.56 C-Source.svobject

© 1996 by Andreas R. Kleinert.

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2

Release Date : 07.06.1996

Description

~~~~~

C-Source.svobject is an external library module for the superview.library.

It creates C-Source code in chunky/RGB notation from any input graphics (<= 8 Bit and HAM as chunky, 24 Bit as RGB).

In detail these are :

Writing :

- 8 Bit as C-Source with UBYTE [] pixel array and UBYTE [][][3] colormap (HAM6/8 indicated by flag #define)
- 24 Bit as C-Source with UBYTE [][][3] pixel array (RGB data)

For simple extraction of a colormap from a graphics file, you should apply the Crop operator with parameters

```
CROP_LEFTEDGE=0
CROP_TOPEDGE=0
CROP_WIDTH=1
CROP_HEIGHT=1
```

and then save the result as C-Source.

Remarks

~~~~~

- this module uses buffered I/O, but nevertheless is quite slow for larger graphics (I/O buffer is set to 64K for OS version >= 39)
- usage of RAM-Disk for output buffers is strongly suggested
- the module does raw output, which should be compileable with any Amiga C compiler - and any other compiler, when simply replacing

```
#include <exec/types.h>
```

with

```
typedef unsigned char UBYTE;
typedef unsigned long ULONG;
```

- the "endmark" entries at the end of any written array can simply be removed; they actually have no meaning, but simplify the process of writing the data to the file

History

~~~~~

V3.2 (07.06.1996) :

- fixed bug in initialization code

V3.1 (16.04.1996) :

- first version: yet only with chunky/RGB support and not very fast

1.57 UtahRLE.svobject

This one is not included with this distribution, but can be found on AmiNet or any related BBS.

Look out for an archive called "svoUtah33.LHA" or similar.

1.58 CyberDataType.svobject

© 1996 by Andreas R. Kleinert.
FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.1
Release Date : 09.02.1996

Description

~~~~~

CyberDataType.svobject supports (CyberGraphX') picture.datatype V43, which upgrades DataTypes to 24 Bit handling.

Ralph Schmidt's picture.datatype V43 does allow loading of 24 Bit pictures by enhancing the datatype system with additional functionality. To be able using these features with SuperView, you need a registered CyberGraphX installation and CyberDataType.

When CyberGraphX and picture.datatype V43+ are both installed in your system, this module will supersede the internal DataType svobject (for graphics only) and thus allow extraction of 8 Bit Chunky and 24 Bit RGB graphics from Picture DataTypes

You need special DataTypes for that (as coming with picture.datatype V43).

### Reading :

- any DataTypes of class "picture" in either 8 Bit (standard)
- and/or 8 and 24 Bit  
(with special datatype modules for picture.datatype V43)

### Remarks

~~~~~

This modules does for example not handle animations or other datatypes, which will still be passed through the standard datatype handler, when the library is configured to handle "ANYDATATYPES".

History

~~~~~

V3.1 (09.02.1996) :

- first version (tested with picture.datatype V43.667)

## 1.59 ECS.svdriver

© 1994-96 by Andreas R. Kleinert.  
FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 2.8

---



Release Date : 21.12.1995

#### Requirements

~~~~~

- OS V2.04+ (V37)+ and its libraries
- superviewsupport.library V5+
- AMIGA with Enhanced Chip Set (ECS)

Description

~~~~~

ECS.svdriver is an external SV\_GfxBuffer display module for superview.library, which allows object-oriented displaying on ECS displays.

This Driver supports the following :

| Dimensions | Depth  | Type                    |
|------------|--------|-------------------------|
| [ECS]      | [ECS]  | BITPLANE                |
| [ECS]      | 8/(24) | ONEPLANE (Chunky Pixel) |

The 8-Bit mode will perhaps only work on ECS systems which have any Graphic Card installed, which allows 256 or more colors in a way of an Intuition emulation.

24 Bit Graphics will be displayed as "best guess of 256 colors" if no SVOperator is specified.

Autoscrolling of Screens larger than the actual display is supported : Just move the mouse to the boundings !

#### ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/ECS.controlpad"
ControlPad-Commands : - 24BITOPERATOR=<OperatorName>
                        ; (case-sensitive, ".svoperator" may be added)
                        ; e.g. "24BITOPERATOR=24BitToHAM.svoperator"
                        ; or   "24BITOPERATOR=ExtractGrayScales"
                        ; specifies, which operation should be performed
                        ; on 24 Bit graphics before displaying them
                        ; (if not specified displaying will fail)
- 8BITOPERATOR=<OperatorName>
  ; (case-sensitive, ".svoperator" may be added)
  ; e.g. "8BITOPERATOR=ExtractGrayScales"
  ; specifies, which operation should be performed
  ; on non-ECS graphics (more than 16 Colors in
  ; HighRes, more than 32 Colors in LowRes, HAM8)
  ; (if not specified, will be tried to display)
- BITMAPCOPY=<DIRECT|RTG>
  ; "BITMAPCOPY=RTG" prevents ECS.svdriver from
  ; directly copying into Bitmaps, which will
  ; result in a usage of more memory, but keeps it
  ; working.
  ; Default is "BITMAPCOPY=DIRECT".
- SCREENINFRONT
  ; Put Screen to front before the graphics
  ; has been displayed (useful with GfxCards)
```

History

~~~~~

V2.8 (21.12.1995) :

- added REV. 2 of new, rock-solid startup code
- fixed several possible crashing reasons
- tried to use LAYERS\_NOBACKFILL with SA\_BackFill and WA\_BackFill even under V37-V38. This caused an odd-address error since the given dummy-pointer to address 1 is only valid for V39 or greater. With V2.7 of ECS.svdriver displaying under V38 and below perhaps was impossible. Fixed.  
(-> reported by Klaus Schneider and Sven Drieling)

V2.7 (29.10.1995) :

- faster closing
- recompiled with SAS/C V6.56

\*\*\*\*\*  
 \* Revison history entries before V2.7 have been deleted for space reasons \*  
 \*\*\*\*\*

**1.60 AGA.svdriver**

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 2.8  
 Release Date : 21.12.1995

## Requirements

~~~~~

- OS V3.00+ (V39)+ and its libraries (takes advantage of V40)
- superviewsupport.library V5+
- AMIGA with AGA ChipSet (ECS still supported, but with restrictions) or an appropriate Graphics Card with Workbench Emulation

Description

~~~~~

AGA.svdriver is an external SV\_GfxBuffer display module for superview.library, which allows object-oriented displaying on AGA displays.

This Driver supports the following :

| Dimensions | Depth  | Type     |                |
|------------|--------|----------|----------------|
| [AGA]      | [AGA]  | BITPLANE |                |
| [AGA]      | 8/(24) | ONEPLANE | (Chunky Pixel) |

24 Bit Graphics will be displayed as "best guess of 256 colors" if no SVOperator is specified.

Autoscrolling of Screens larger than the actual display is supported :  
Just move the mouse to the boundings !

#### ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/AGA.controlpad"
ControlPad-Commands : - 24BITOPERATOR=<OperatorName>
                      ; (case-sensitive, ".svoperator" may be added)
                      ; e.g. "24BITOPERATOR=24BitToHAM.svoperator"
                      ; or   "24BITOPERATOR=ExtractGrayScales"
                      ; specifies, which operation should be performed
                      ; on 24 Bit graphics before displaying them
                      ; (if not specified, "best guess" colors will
                      ; be used, which is really slow)
- BITMAPCOPY=<DIRECT|RTG>
  ; "BITMAPCOPY=RTG" prevents AGA.svdriver from
  ; directly copying into Bitmaps, which will
  ; result in a usage of more memory, but keeps it
  ; working.
  ; Default is "BITMAPCOPY=DIRECT".
- SCREENINFRONT
  ; Put Screen to front before the graphics
  ; has been displayed (useful with GfxCards)
```

History

~~~~~

V2.8 (21.12.1995) :

- added REV. 2 of new, rock-solid startup code
- fixed several possible crashing reasons

V2.7 (29.10.1995) :

- faster closing
- recompiled with SAS/C V6.56

```
*****
* Revison history entries before V2.7 have been deleted for space reasons *
*****
```

## 1.61 CyberGraphics.svdriver

© 1995-96 by Andreas R. Kleinert.

FREWARE. All rights reserved. Only to be distributed with superview.library.

```
Version      : 2.8
Release Date : 10.02.1996
```

#### Requirements

~~~~~

- OS V3.00+ (V39)+ and its libraries
- 68020 or greater processor

- superviewsupport.library V6+
- cybergraphics.library V40+

- AMIGA with CyberGraphX System and appropriate Graphics Card

Description

~~~~~

CyberGraphics.svdriver is an external SV\_GfxBuffer display module for superview.library, which allows object-oriented displaying on Amigas with the 16/24 Bit CyberGraphX Workbench Emulation System.

This Driver displays any 1..8 or 24 Bit graphics via the CyberGraphX System. Graphics upto 256 Colors will be displayed via usual graphics routines, 24 Bit graphics will be displayed via CyberGraphX in either 16 or 24 Bit (16 Bit is default, because the unregistered version of CyberGraphX does not support more and I guess, that not yet all the users out there registered it already).

Currently BITPLANE with more than 8 Bit Colordepth and ONEPLANE with other pixelbits than 8 or 24 (e.g. 16 Bit R5:G5:B5:1) are not supported. Also EHB graphics may not result in correct displays.

HAM6/8 will be converted to 24 Bit, which then will either be displayed as 16 or 24 Bit.

The driver will not open any Screens in standard ECS/AGA modes, since it makes more sense to use AGA.svdriver in such a case. If no CyberGraphX screenmode is specified, it will be generated, which is just the same effect like with an intelligent screen promoter. No ECS/AGA screenmodes will be passed through, since this is neither useful nor practicable on two Monitor systems or with single 31khZ-limited Monitors.

#### Credits

~~~~~

Thanks to Ingenieurbüro Helfrich, for supplying the PiccoloSD64 card.

The CyberGraphX Software is of course copyrighted by its authors, which is hereby expressively respected in all points.

ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/Cybergraphics.controlpad"
ControlPad-Commands : - EMUSCREENDEPTH=<16|24>
                      ; beginning depth for opening CyberGraphX Screens
                      ; Default is 16 Bit (which will also be tried
                      ; when opening of a 24 Bit Screen fails)
                      ; This does not concern colordepths < 16 Bit,
                      ; except HAM6/8.
- SMALLSCREENS
  ; unless this KeyWord is specified, it is not
  ; tried to open screens smaller than 320x240
```

## History

~~~~~

V2.8 (10.02.1996) :

- new C startup-code was compiled for 68020. Fixed.
- this revision history was slightly wrong

V2.7 (13.01.1996) :

- added REV. 2 of new, rock-solid startup code
- using new GST
- now requesting superviewsupport.library V6+
- did distort any 8 Bit odd width graphics (not divideable by 16 without rest) when displaying under OS V3.00 (V39). This did not happen with graphics < 320x240 and the SMALLSCREENS option unset and it did not concern OS V3.1 (V40) users. (-> Jürgen Schäfer)
May now be slightly slower in displaying such graphics, since these will be displayed line-wise. Aligned graphics will be displayed using the old method.

V2.6 (29.10.1995) :

- faster closing
- recompiled with SAS/C V6.56

 * Revison history entries before V2.6 have been deleted for space reasons *

1.62 EGS7.svdriver

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 2.3

Release Date : 29.10.1995

Requirements

~~~~~

- OS V2.04+ (V37)+ and its libraries
- superviewsupport.library V5+
- egs.library V1+ (should be V7+)
- egsintui.library V1+ (should be V7+)
- egsgfx.library V1+ (should be V7+)
- AMIGA with EGS Graphic Card (or EGS distribution with Amiga Emulation)

## Description

~~~~~

EGS7.svdriver is an external SV_GfxBuffer display module for superview.library, which allows object-oriented displaying on EGS V7 (Enhanced Graphic System).

This Driver displays any 1..8 or 24 Bit graphics via the EGS7

System. Any source Graphics with less than 24 Bit, like 256 Color or HAM6/8 graphics, are converted to 24 Bit before displaying them.

Currently BITPLANE with more than 8 Bit Colordepth and ONEPLANE with other pixelbits than 8 or 24 (e.g. 16 Bit R5:G5:B5:1) are not supported. Also EHB graphics may not result in correct displays.

This EGS7.svdriver is based on my former "EGS.svdriver", which is to be replaced by the new one. But of course you may still continue using the older one.

The difference is, that the EGS7.svdriver really does display any of the available source graphics formats (even HAM) and is x-times faster than the old one. Also, it is now possible to directly close the EGS-Displaywindow via its Close-Gadget, since some kind of self-written Intuition-Emulation enables us to do so.

Credits

~~~~~

Thanks to Ingenieurbüro Helfrich, for supplying the PiccoloSD64 card.

The EGS7 Software is of course copyrighted by its authors, which is hereby expressively respected in all points.

#### Notes

~~~~~

This EGS-SVDriver should still work with the AMIGA driver for EGS.

When using the ECS/AGA emulation, you should set the max. possible ColorDepth in the ScreenMode preferences program.

AGA:

With AGA any Graphics with less than 256 Colors should be displayed 100% correctly. Only 256 Color-graphics will usually have some Colors wrong, because those are obtained by the EGS-System for the Display itself (Window-Borders, etc.), so that they usually can't be used for the graphics. 24 Bit graphics will be dithered to 256 Colors under AGA (usually GreyScaled).

ECS:

Using the ECS emulation will perhaps nearly always result in very ugly Colors, if you're displaying more than, let's say, 8 Colors. This results out of the maximum ColorDepth of 16 Colors in Hires, of which some - see AGA notes - are already reserved. Of course 24 Bit graphics may also be dithered to 16 Colors/GreyScales, but better don't try it out ...

History

~~~~~

V2.9 (29.10.1995) :

- recompiled with SAS/C V6.56

V2.2 (9.10.1995) :

---

- fixed version counting for info structure

V2.1 (28.7.1995) :

- rewrote "EGS.svdriver" and called it "EGS7.svdriver" now
- is x-times faster now
- displays any available GfxBuffer format (even HAM) in 24 Bit now
- uses kind of self-made Intuition Emulation now, no longer a requester-window on the Workbench has to be used to close the EGS-Screen, but instead using the EGS-Window's Close-Gadget suffices now (also the most important IDCMP actions, like IDCMP\_MOUSEBUTTONS, -RAWKEY, -VANILLAKEY and -CLOSEWINDOW are already supported). Suffices e.g. for SuperView.

## 1.63 Picassoll.svdriver

© 1995-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 2.3  
Release Date : 29.10.1995

### Requirements

~~~~~

- OS V2.04+ (V37)+ and its libraries
- superviewsupport.library V5+
- vilintuisup.library V1+
- AMIGA with Picasso II Graphics Card and Software

Description

~~~~~

PicassoII.svdriver is an external SV\_GfxBuffer display module for superview.library, which allows object-oriented displaying on Amigas with the Picasso Workbench Emulation System.

The corresponding Picasso-libraries are needed.

This Driver supports the following :

| Source   | Depth | Special Modes | Dimensions           | Dest. Depth |
|----------|-------|---------------|----------------------|-------------|
| ONEPLANE | 8/24  | HAM6, HAM8    | [ Picasso-Dependent] | 8 / 24      |
| BITPLANE | 1..8  | HAM6, HAM8    | [ Picasso-Dependent] | 8 / 24      |

Currently BITPLANE with more than 8 Bit Colordepth and ONEPLANE with other pixelbits than 8 or 24 (e.g. 16 Bit R5:G5:B5:1) are not supported. Also EHB graphics may not result in correct displays.

HAM6/8 will be converted to 24 Bit, which then will either be displayed as 16 or 24 Bit.

24 Bit will either be displayed in 16 or 24 Bit.

### Credits

~~~~~

Thanks to Florian Zeiler (Irseesoft) for supplying the Gfx Card.

The Picasso Emulation Software is of course copyrighted by Village Tronic Marketing GmbH, which is hereby expressively respected in all points.

Known Bugs

~~~~~

##### Problem:

As with V1.8 (6. Apr 1994) of the Picasso II Software Set, the vilintuisup.library sometimes supplies Screens, which differ between internal dimensions and actual dimensions. A 24 Bit graphics with 1165x712 would be displayed on a Screen with correct dimensions in best ViewMode. But for some reason, the Intuition Part of the Screen will be opened in 1168x712 with a related ViewMode. Whenever this ViewMode is the best fitting one, this seems to make problems to the Picasso Emulation. If the ViewMode is one with a larger or smaller display (so that either Autoscroll has to be used, or the graphics does not fill the whole screen) this does not occur. This is not a bug of the PicassoII.svdriver, since width adjustments will occur independently from the viewmode selection.

##### Solution:

Whenever the Picture is weirdly distorted, at first try an other viewmode. This bug does not seem to cause crashes, only destroyed displays. Maybe a newer version of the Picasso Software already did fix this (seems to be an alignment problem when copying from RAM into the Picasso onboard-memory).

##### Problem:

When using AUTOSCROLLADJUST the Picasso Software may destroy about two lines within the displayed picture (get black).

##### Solution:

Don't use it.

#### ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/PicassoII.controlpad"
ControlPad-Commands : - SMALLSCREENS
                      ; unless this KeyWord is specified, it is not
                      ; tried to open screens smaller than 320x240
- BLITTER
                      ; uses the Picasso blitter to copy graphics
                      ; into screens. Otherwise CPU is used.
                      ; On 68000 systems you may wish to use the blitter
- EMUSCREENDEPTH=<16|24>
                      ; beginning depth for opening Picasso Screens
                      ; Default is 24 Bit (if opening fails, it is
                      ; also tried to open a 16 Bit Screen, then)
- SCREENMODEREQUEST
                      ; if this one is specified, the SVDriver ITSELF
                      ; will open a ScreenMode-Requester and ask
                      ; for an appropriate Screenmode to use.
                      ; Useful, if you always like to change modes.
- AUTOSCROLLADJUST
```



```

; this keyword will force Autoscroll whenever
; it would make sense, but the Picasso Software
; would not manage it by itself (when either
; only width or height need to be autoscrollled)

```

History

~~~~~

V2.3 (29.10.1995) :

- recompiled with SAS/C V6.56

```

*****
* Revison history entries before V2.3 have been deleted for space reasons *
*****

```

## 1.64 OPAL.svdriver

Just a short note before using and installing OPAL.svdriver and (of course ;- ) reading the following documentation:

There's a little problem causing some OPAL programs not to be displayed correctly on certain systems with specific chipsets.

That was the reason why it did not work on the machines of some people - mine was one of these - but nevertheless it is easy to workaround and even already has been described within the official OpalVision Reference Manual [page 326].

To speak clearly, the color palette has to be adjusted concerning Color 0 (the magic color), but only the BLUE component, within the Palette Preferences. Any other colors and components are freely selectable.

OCS and ECS Chipset Machines with Zorro Bus (2000-3000)

BLUE of COLOR 0 must be either 1, 3, 5, 7, 9, 11, 13, or 15.

AGA Chipset Machines with Zorro Bus (4000)

BLUE of COLOR 0 must have Bit 4 set, which restricts it to one of the following ranges:

```

          16-31   or   48-63   or   80-95   or   112-127
or      144-159 or  176-191 or  208-223 or  240-255

```

Otherwise you may get a black or weirdly distorted screen (I tested it ;- ) or must use some tricks and Amiga+M fiddling to get a picture.

```

*****
included the original documentation at this place
*****

```

OPAL.svdriver/Documentation

OPAL.svdriver/Documentation

## PROGRAM

OPAL.svdriver  
Version 2.3  
Release date: 29.6.95

## COPYRIGHT

© 1995 Paul Huxham and Steve Quartly.  
Bonusware, all rights reserved.

## DISTRIBUTION

Freely distributable with any non-commercial application that uses superview, however all copyright remains with the authors. This distribution archive may not be split for further distribution.

## PURPOSE

OPAL.svdriver is an external display driver module for superview.library. It can display superview buffers on an OpalVision display card.

## CONTENTS

OPAL.svdriver - This is the driver to copy to LIBS:svdrivers.  
OPAL.svdriver.doc - Documentation you are reading now.

## USAGE

Select OPAL.svdriver as the display driver from SuperViewPrefs and/or from the superview compatible software you are using.

While conversion for OpalVision is taking place, the following information will be displayed:

OPAL.svdriver version number,  
and the image size (which can be larger than the screen size).

## REQUIREMENTS

- 1) An OpalVision card :-)
- 2) opal.library must be in LIBS: otherwise the driver will fail with an internal error.
- 3) A minimum of Kickstart 2.0.

## CONTROL PAD

ControlPad-Name:  
"ENV:superview-library/OPAL.controlpad"

## ControlPad-Commands:

OVERSCAN=ON|OFF

Force Opal to display in either overscan or non-overscan. Not specifying an overscan setting informs OPAL.svdriver to intelligently select an overscan screen mode for you.

CENTER\_IMAGE

Force the displayed image to be taken from the centre section of the image buffer, i.e. If the image is 640 x 512 and the display mode is LoRes, Non-interlace and CENTER\_IMAGE is specified then the top left of the visible display will be 160, 156 offset into the original image.

If not present, the image will be displayed from the top left of the image buffer.

## NOTES

Opal images cannot be scrolled yet (until superview supports scrolling images larger than the drivers screen size), so if the picture is larger than the display area, the displayed image will be cropped.

#### HISTORY

V0.0 First the earth cooled.

V1.0 Initial release.

V2.2 Supports new features of superview.library version 11.6

- Selectable screen modes.
- Control pads for configuration.

V2.3 - Fixed byte aligning of 24 bit images.

- Drastically reduced memory requirements and increased speed by removing the conversion from RGB triplets through RGB planes.

#### COMPLIER

OPAL.svdriver was written using CED V3.5 and compiled with SAS/C 6.55 on an Amiga 2000/030 and Amiga 4000/040. Enforcer was used to detect and correct programming errors.

#### BUGS

Should you find any bugs, please report them so that they can be fixed. Likewise any suggestions for improvement of the driver should be forwarded so that they can be addressed.

#### THANKYOU

Very many thanks to Andreas, the author of Superview.library for implementing some of our suggestions and also for explaining some of the internals of superview.library. Without superview this driver would be a boat anchor. (It would probably float :-)

#### AUTHORS

You can contact the authors via:

Email:

Paul Huxham  
paulh@Perth.DIALix.oz.au

Steve Quartly  
steveq@sndcrft.DIALix.oz.au

or

P.O. Box 875  
Morley,  
Perth,  
Western Australia 6943

## 1.65 Retina.svdriver

© 1995-96 by Andreas R. Kleinert.

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 2.8

Release Date : 29.10.1995

---

## Requirements

~~~~~

- OS V2.04+ (V37)+ and its libraries
- superviewsupport.library V6+
- retina.library V7+
- retinaemu.library V1+
- AMIGA with Retina Z2/Z3 Graphics Card and Software

Description

~~~~~

Retina.svdriver is an external SV\_GfxBuffer display module for superview.library, which allows object-oriented displaying on Amigas with the 16/24 Bit Retina Workbench Emulation System.

The corresponding Retina-libraries are needed.

This Driver supports the following :

| Source   | Depth | Special Modes | Dimensions          | Dest. Depth |
|----------|-------|---------------|---------------------|-------------|
| ONEPLANE | 8/24  | HAM6, HAM8    | [ Retina-Dependent] | 16 / 24     |
| BITPLANE | 1..8  | HAM6, HAM8    | [ Retina-Dependent] | 16 / 24     |

Currently BITPLANE with more than 8 Bit Colordepth and ONEPLANE with other pixelbits than 8 or 24 (e.g. 16 Bit R5:G5:B5:1) are not supported. Also EHB graphics may not result in correct displays.

HAM6/8 will be converted to 24 Bit, which then will either be displayed as 16 or 24 Bit.

## USAGE

~~~~~

RetinaEmu:

For SuperView's "svdRetina-Screen" screen entry, select an AutoScroll-Area of "800x600" and "No Refresh" with "Retina Standard" (Z2) or "Retina Blitter" (Z3) output.

RetinaScreenMode:

The Monitor specifications need to be 100% correct.

What using instead?

Retina.svdriver always displays graphics in 16/24 Bit - so when running on AGA systems, AGA.svdriver would be the better solution for upto 256 Colors. On ECS systems this may only apply to 16/32 Color graphics as long as you don't have OS 3.1 and a RetinaEmu which supports 256 Color Screens for this configuration (should be available). Otherwise with ECS.svdriver you would still have to specify "8BITOPERATOR" for more than 16/32 Colors, which then would be as slow or even slower.

Credits

~~~~~

Thanks to Florian Zeiler (Irseesoft) for supplying the Gfx Card.

The Retina Emulation Software is of course copyrighted by

MS MacroSystem Computer GmbH Germany, which is hereby expressively respected in all points.

#### ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/Retina.controlpad"

ControlPad-Commands : - EMUSCREENDEPTH=<16|24>

; beginning depth for opening Retina Screens
; Default is 24 Bit (if opening fails, it is
; also tried to open a 16 Bit Screen, then)

History

~~~~~

V2.8 (29.10.1995) :

- recompiled with SAS/C V6.56

V2.7 (9.10.1995) :

- fixed version counting for info structure

V2.6 (1.10.1995) :

- COMPLETE, MAJOR REWORK.
- fixed several bugs; did some thousand workarounds %-)
- removed several controlpad entries :-)
- fixed width, height and bpl parameters for cases, where screensize and buffersize do differ (larger or smaller than expected/requested/desired)
- fixed screenmode generation for BestModeID generation with graphics, where width<height. Before, always NTSC:LowRes was used (returned by OS function), which caused crashes. Side effect sometimes are better x/y aspect results.
- changed way of autoscroll activation (as for Retina.svdriver it's always active, but RetinaEmu needs to be informed about)
- now always allows to select and specify ScreenModes, as long as compliant to the needs of the Retina (when being passed, they'll be checked. Valid ViewModes also are available about the ViewMode-Requester when using the list supplied by Retina.svdriver, e.g. fromout SuperView)
- requests support-lib V6+ now

\*\*\*\*\*  
\* Revison history entries before V2.6 have been deleted for space reasons \*  
\*\*\*\*\*

## 1.66 PICASSO.svdriver

This one is a "third party product", which has been written by Patrick Ohly.

It is not included with this distribution, but can be found on AmiNet or any related BBS.

Look out for an archive called "SVDPicasso.LHA" or similar.  
On AmiNet:

V1.0: SVDPicasso.LHA  
V2.0: SVDPicasso2\_0.LHA

Any of the official SuperView Support BBSs should also supply a copy.

## 1.67 MERLIN.svdriver

This one is a "third party product", which has been written  
and is © 1995 by Thomas Eigentler.

Version : 2.2  
Release Date : 28.07.1995

### Description

~~~~~

Merlin.svdriver is an external SV_GfxBuffer display module for
superview.library, which allows object-oriented displaying on
Amigas with the 16/24 Bit Merlin Workbench Emulation System.

The corresponding Merlin-libraries are needed.

Upto V11.8 it had been distributed separately, e.g. on AmiNet, but
as with V11.9 it has been included into the superview.library
distribution with friendly permission by the author.

Author

~~~~~

For any suggestions, bug-reports and ideas you should directly  
contact Thomas Eigentler, via one of his email addresses:

FidoNet: Thomas Eigentler 2:246/1511.0  
UseNet: Thomas\_Eigentler@damage.tynet.sub.org  
thomas.eigentler@student.uni-tuebingen.de

### Credits

~~~~~

The Merlin Emulation Software is of course copyrighted by it's
authors, which is hereby expressively respected in all points.

History

~~~~~

Here's a short overview of the previous releases of the Driver  
(concatenated and translated by me, not Thomas - so don't rely on it ;-)

V2.2 (28.07.1995):

---

- now also can handle ONEPLANE HAM-Graphics  
(was the reason why Merlin.svdriver always did crash during  
24Bit/HAM conversion. Works now.)
- checks, whether the systems actually contains a Merlin Graphics  
(it is checked, whether the hrgsystem.library is available.  
Versions <41 return a NULL pointer with inactive WB-Emu.  
The current version (>41) is able to be opened even without  
WB-Emu, so that Merlin-Driver will then be selectable, but  
can't open a screen, though.  
Will only be relevant for Merlin-Users, anyhow. It would be  
possible to test the WB-Emu via opening any screen, but  
that wouldn't be very useful).
- is a little bit faster now

#### V2.0:

- shipped with beta-releases of superview.library 11.9
- includes 68000 and 68020 versions

#### V1.01:

- up to superview-rules Merlin Driver is now released as SVD\_MerlinDriver.
- some 24-Bit pictures crashed - fixed.
- support of HAM8/HAM6 pictures
- problem with BytesPerRow - fixed.

#### V1.00:

- first public release

## 1.68 XOR.svoperator

© 1994-96 by Andreas R. Kleinert.

FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2  
Release Date : 29.10.1995

#### Description

~~~~~

XOR.svoperator is an external SV_GfxBuffer modification module
for superview.library V9+.

XOR.svoperator modifies a graphics in a way, that the color register
numbers of the supplied picture's pixels are XOR'ed with a given value,
which may be specified via ControlPads (see below).

ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/XOR.controlpad"
ControlPad-Commands : - XORVALUE=<0..255>
                      ; 0 and 255 will not be the best decision ;-)
```

```
                      ; Use 15 or something like this.
```

#### History





\*\*\*\*\*

## 1.70 Crop.svoperator

© 1995-96 by Andreas R. Kleinert.

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2  
Release Date : 29.10.1995

### Description

~~~~~

Crop.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

Crop.svoperator may extract shapes of any size from any 2..256 Color or 24 Bit graphics.

Values for "LeftEdge", "TopEdge", "Width" and "Height", describing the crop box have to be supplied via ControlPad settings.

ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/Crop.controlpad"
ControlPad-Commands : - CROP_LEFTEDGE=<Value>
                      ; crop from x position
                      ; (will be adjusted, if >= source width)
                      - CROP_TOPEdge=<Value>
                      ; crop from y position
                      ; (will be adjusted, if >= source height)
                      - CROP_WIDTH=<Value>
                      ; crop how many x pixels from leftedge
                      ; (will be adjusted, if too large)
                      - CROP_HEIGHT=<Value>
                      ; crop how many y pixels from topedge
                      ; (will be adjusted, if too large)
```

### History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

* Revison history entries before V3.1 have been deleted for space reasons *

1.71 Dither24Bit.svoperator

© 1994-96 by Andreas R. Kleinert.
 (Also see notes under "Credits".)
 FREeware. All rights reserved. Only to be distributed with superview.library.

Version : 3.2
 Release Date : 29.10.1995

Description

~~~~~

Dither24Bit.svoperator is an external SV\_GfxBuffer modification module for superview.library V9+.

Dither24Bit.svoperator dithers 24 Bit RGB graphics to 256 Colors by default. Possible is also dithering to less colors (2..128), if specified via ControlPad-Settings.

This module applies Heckbert's median cut and dithers using Floyd-Steinberg.

#### ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/Dither24Bit.controlpad"
 ControlPad-Commands : - COLORDEPTH=<1..8>
 ; specifies the colordepth of the dithering
 ; output (1->2 Colors .. 8->256 Colors)

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (09.10.1995) :

- upgraded to V2/V3 specs
- now uses black&white for 2 color dithering mode (-> Florian Zeiler)

\*\*\*\*\*  
 \* Revison history entries before V3.1 have been deleted for space reasons \*  
 \*\*\*\*\*

## 1.72 HilbertDither256.svoperator

© 1994-96 by Andreas R. Kleinert.  
 (Also see notes under "Credits".)  
 FREeware. All rights reserved. Only to be distributed with superview.library.

Version : 3.2  
 Release Date : 29.10.1995

#### Description

~~~~~

HilbertDither256.svoperator is an external SV_GfxBuffer modification module

for superview.library V9+.

HilbertDither256.svoperator modifies any supplied source SV_GfxBuffer in a way, that all supplied graphics with upto 256 Colors are dithered to Black & White graphics (2 Colors), as e.g. needed for desktop publishing or output on matrix printers.

It uses the fractal Hilbert curve for getting best results in eliminating the resulting errors.

As a side effect, the resulting picture will always have a width and height, which is divideable by 16 (graphics will be adjusted this way).

ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/HilbertDither256.controlpad"
ControlPad-Commands : - BACKGROUND=<BLACK|WHITE>
                      ; defines, which of the two colors will act
                      ; as background color. Useful e.g. for printing.
```

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

1.73 AnyTo24Bit.svoperator

© 1995-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

```
Version      : 3.2
Release Date : 29.10.1995
```

Description

~~~~~

AnyTo24Bit.svoperator is an external SV\_GfxBuffer modification module for superview.library V9+.

AnyTo24Bit.svoperator converts any input graphics into TrueColor RGB graphics with 24 Bit.  
Input may be any (upto) 256 Color graphics or HAM6/HAM8 data.

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs
 - HAM8 -> 24 Bit conversion had not been fixed in 1.6
 Now it has.
 (-> Florian Zeiler, Jürgen Schäfer)

 * Revison history entries before V3.1 have been deleted for space reasons *

1.74 ExtractGrayScales

© 1994-96 by Andreas R. Kleinert.
 FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.4
 Release Date : 21.12.1995

Description

~~~~~

ExtractGrayScales.svoperator is an external SV\_GfxBuffer modification module for superview.library V9+.

ExtractGrayScales.svoperator modifies any supplied source SV\_GfxBuffer in a way, that its colormap will be changed to reflect gray-scales (works with (upto) 256 Colors and 24 Bit Files).  
 Output is done in input colordepth or 256 Colors by default (fastest).

This is a sample SVOperator for simple demonstration how to write one. It's simple, but effective.

ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/ExtractGrayScales.controlpad"
 ControlPad-Commands : - COLORDEPTH=<1..8>
 ; specifies the colordepth of the grayscale
 ; output (1->2 Colors .. 8->256 Colors)
 - QUICK
 ; uses >>2, >>1, >>3 (*0.25, *0.5, *0.125)
 ; instead of *0.3, *0.59, *0.11

History

~~~~~

V3.4 (21.12.1995) :

- added REV. 2 of new, rock-solid startup code

V3.3 (16.12.1995) :

- added new, rock-solid startup code  
 - no longer causes crashes  
 - now using new GST

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

## 1.75 ExtractRed

© 1994-96 by Andreas R. Kleinert.

FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2  
Release Date : 29.10.1995

Description

~~~~~

ExtractRed.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

ExtractRed.svoperator modifies any supplied source SV_GfxBuffer in a way, that only the RED values of a picture will be extracted for creation of a new (upto) 256 Color graphics (works with (upto) 256 Colors and 24 Bit Files).

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

## 1.76 ExtractGreen

© 1994-96 by Andreas R. Kleinert.

FREEMWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2  
Release Date : 29.10.1995

## Description

~~~~~

ExtractGreen.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

ExtractGreen.svoperator modifies any supplied source SV_GfxBuffer in a way, that only the GREEN values of a picture will be extracted for creation of a new (upto) 256 Color graphics (works with (upto) 256 Colors and 24 Bit Files).

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

\*\*\*\*\*  
 \* Revison history entries before V3.1 have been deleted for space reasons \*  
 \*\*\*\*\*

## 1.77 ExtractBlue

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2

Release Date : 29.10.1995

## Description

~~~~~

ExtractBlue.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

ExtractBlue.svoperator modifies any supplied source SV_GfxBuffer in a way, that only the BLUE values of a picture will be extracted for creation of a new (upto) 256 Color graphics (works with (upto) 256 Colors and 24 Bit Files).

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

\*\*\*\*\*  
 \* Revison history entries before V3.1 have been deleted for space reasons \*  
 \*\*\*\*\*

\*\*\*\*\*

## 1.78 TopToBottom

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2  
Release Date : 29.10.1995

### Description

~~~~~

TopToBottom.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

TopToBottom.svoperator modifies any supplied source SV_GfxBuffer in a way, that it will be swapped from bottom to top (works with (upto) 256 Colors and 24 Bit Files).

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

\*\*\*\*\*  
\* Revison history entries before V3.1 have been deleted for space reasons \*  
\*\*\*\*\*

## 1.79 LeftToRight

© 1994-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2  
Release Date : 29.10.1995

### Description

~~~~~

LeftToRight.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

LeftToRight.svoperator modifies any supplied source SV_GfxBuffer in a way, that it will be swapped from left to right (mirrored) (works with (upto) 256 Colors and 24 Bit Files).

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

## 1.80 Rotate

© 1995-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2  
Release Date : 29.10.1995

Description

~~~~~

Rotate.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

Rotate.svoperator rotates any (upto) 256 Color or 24 Bit graphics by (default) 90 degrees (reverse clockwise).

The default behaviour may be overwritten via controlpad settings, which also allow 180 and 270 degrees (no extra memory needed: uses different algorithm).

ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/Rotate.controlpad"

ControlPad-Commands : - DEGREES=<90|180|270>

                  ; rotate by how many degrees (reverse clockwise) ?

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

```
*****
* Revison history entries before V3.1 have been deleted for space reasons *
*****
```

1.81 RotateFree

© 1995-96 by Andreas R. Kleinert.
 (Also see notes under "Credits".)

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.4
 Release Date : 13.01.1996

Description

~~~~~

RotateFree.svoperator is an external SV\_GfxBuffer modification module for superview.library V9+.

Rotate.svoperator rotates any (upto) 256 Color or 24 Bit graphics by 0..360 (default is 90) degrees (reverse clockwise) about any given point (default is the middle, which is width/2, height/2). The default behaviour may be overwritten via controlpad settings.

#### ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/RotateFree.controlpad"
ControlPad-Commands : - ROTATE_ANGLE=<0..360>
                      ; rotate by how many degrees (reverse clockwise) ?
                      - METHOD=<MIDDLE|GIVENPOINT>
                      ; default is rotation about the middle
                      - X_COORD=<Value>
                      ; if METHOD=GIVENPOINT we need (X/Y)
                      ; Default is (0/0)
                      - Y_COORD=<Value>
                      ; if METHOD=GIVENPOINT we need (X/Y)
                      ; Default is (0/0)
                      - KEEPSIZE
                      ; if this keyword is set, it is NOT tried to
                      ; readjust the image size and to center the
                      ; image, so that parts, which would be out of
                      ; range won't be cut off (which works best
                      ; with (X/Y) pairs in the left, upper quarter)
```

History

~~~~~

V3.4 (13.1.1996) :

- mathtrans.library was not opened by the code, but AUTOOPENLIB instead -> did crash ALWAYS
- it seems as if the opening (can't say, whether the libraries themself) of transient math libraries (like "mathtrans.library" together with "mathffp.library" or "mathieeedoubtrans.library" together with "mathieeee.library") did cause all the concerned SVObjects and/or SVOperators to crash on certain systems like 68060, some 68000 and some 68040 (on my system only, when running SnoopDOS at the same time to catch library openings).

No problems when only using "mathffp.library" or using link libraries for both, non-transient and transient functions.  
 No more crashes.

We now do that.

- added correct sscanf() routine (.lib)

V3.3 (21.12.1995) :

- added REV. 2 of new, rock-solid startup code

V3.2 (16.12.1995) :

- added new, rock-solid startup code
- replaced IEEE with FFP support (due to crashes)
- no longer causes crashes
- now using new GST

V3.1 (29.10.1995) :

- first version

## 1.82 Scale50

© 1995-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.2

Release Date : 29.10.1995

Description

~~~~~

Scale50.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

Scale50.svoperator scales any (upto) 256 Color or 24 Bit graphics to their half size by default.

The default behaviour may be overwritten via controlpad settings, which also allows to double the size instead.

ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/Scale50.controlpad"

ControlPad-Commands : - METHOD=<HALF|DOUBLE>  
; scale to which size ?

History

~~~~~

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

* Revison history entries before V3.1 have been deleted for space reasons *

1.83 CallPNM

© 1995-96 by Andreas R. Kleinert.

FREEWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.1
Release Date : 07.06.1996

Description

~~~~~

CallPNM.svoperator is an external SV\_GfxBuffer modification module for superview.library V9+.

This is not an operator in the common sense.

Instead it is just an Interface to allow easy access to any external operators from the well known PBM package (distributed as NetPBM).

When using this operator, the given GfxBuffer will be saved as a 24 Bit PNM file (8 Bit input as well), after that the specified PBM program module will be externally called and its output will be written into another temporary file.

This file then will be parsed through superview.library (perhaps it will be in PNM format, but one never know) and the buffer will be returned as the result of the "CallPNM" operation.

Please note, that the whole internal construction of the operator is a little bit more complicated than usual, so that any error handling will perhaps not always result in very clear statements (can't parse the output of the PBM programs).

BTW, any temporary files will of course be placed into "VMEM:", from where they will be deleted later.

### ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/CallPNM24.controlpad"
ControlPad-Commands : - PNMCOMMAND=<PBM command path and name>
                       ; PNM command to be called
                       ; e.g. PNMCOMMAND=Work:NetPBM/pnm/pnmscale
                       - PNMOPTIONS=<Options>
                       ; Options for PNM command to be called
                       ; e.g. PNMOPTIONS=-xscale 2.0 -yscale 2.0
```

History

~~~~~

V3.1 (07.06.1996) :

- there was a little bit messed up concerning version information
- NOW the module has been:
  - recompiled with SAS/C V6.56
  - upgraded to V2/V3 specs
- added new, rock-solid startup code
- PNM command now called in double quotes

\*\*\*\*\*

\* Revision history entries before V3.1 have been deleted for space reasons \*  
 \*\*\*\*\*

## 1.84 OptimizePalette

© 1995-96 by Andreas R. Kleinert.

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.3  
 Release Date : 08.01.1996

Description

~~~~~

OptimizePalette.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

OptimizePalette.svoperator filters all unused colors out of a palette of a given graphics with upto 256 Colors (EHB, HAM6/8 and 24 Bit graphics will be rejected).

A new palette will be created - which also misses any duplicate color table entries - to which then the given graphics is remapped.

This Operator may have three effects (plus combinations):

- a) none, if the palette already was or has been optimized
- b) only some black colors at the end of the color map, if removing of unused colors did not suffice to fall back to the next lower depth boundary (≤ 256 to $\leq 1, 2, 4, 8, 16, 32, 64$ or 128)
- c) a reduced color depth to one of the next lower depth boundaries (as described under b) if there've been a lot of actually unused colors
- d) combinations of b) and c)

History

~~~~~

V3.3 (8.1.1996) :

- added REV. 2 of new, rock-solid startup code
- now using new GST
- the svoperator shifted the image left by one pixel, adding garbage to the right border, when color reduction was performed. (-> Henrik Tikanvaara)

V3.2 (29.10.1995) :

- recompiled with SAS/C V6.56

V3.1 (08.10.1995) :

- upgraded to V2/V3 specs

\*\*\*\*\*  
 \* Revision history entries before V3.1 have been deleted for space reasons \*

\*\*\*\*\*

## 1.85 PaletteDither.svoperator

© 1996 by Andreas R. Kleinert.

(Also see notes under "Credits".)

FREWARE. All rights reserved. Only to be distributed with superview.library.

Version : 3.1  
Release Date : 27.01.1996

### Description

~~~~~

PaletteDither.svoperator is an external SV_GfxBuffer modification module for superview.library V9+.

PaletteDither.svoperator dithers 24 Bit RGB graphics 2..256 color graphics by using a selectable number of colors from a specified fixed palette for that (taken from a given graphics file) and calculating the (optional) rest by itself.

This is useful for creation of ANIMations, where the palette for all single frames have to be (mostly) identical or for games and other applications where the palette of used graphics have to harmonize together.

ControlPad-Switches

~~~~~

```
ControlPad-Name      : "ENV:superview-library/PaletteDither.controlpad"
ControlPad-Commands : - COLORDEPTH=<1..8>
                      ; specifies the colordepth of the final,
                      ; dithered graphics (1..8 for 2..256 colors)
- PALETTEDEPTH=<1..8>
                      ; number of colors (depth) to be taken from the
                      ; fixed palette - the possible rest will be
                      ; generated (if specifying more palette colors
                      ; than available, all available will be taken)
- PALETTEFILE=<palette graphics filename>
                      ; any 2..256 color graphics file, of which
                      ; superview.library is able to extract a
                      ; palette from (for example an IFF-ILBM file,
                      ; but including a BMHD and - maybe empty - BODY).
                      ; Allows to take one graphics as sample for
                      ; the others (concerning the palette).
```

### History

~~~~~

V3.1 (27.01.1996) :

- first version
- (-> Michael Flad)

1.86 Literature used during development of this program

Literature:

~~~~~

- [ 1] "Bitmapped Graphics", 2nd Edition, Steve Rimmer, Windcrest/McGraw-Hill,  
© 1993 by Windcrest Books (registered Trademark of TAB Books).  
ISBN 0-8306-4209-9
- [ 2] "Supercharged Bitmapped Graphics", Steve Rimmer, Windcrest/McGraw-Hill,  
© 1992 by Windcrest Books (registered Trademark of TAB Books).  
ISBN 0-8306-3788-5
- [ 3] "Das Handbuch der Grafikformate", Klaus Holtorf,  
© 1994 Franzis-Verlag GmbH, München  
ISBN 3-7723-6392-X
- [ 4] "Amiga Magazin", Issue 2/1992, Markt & Technik Verlag AG
- [ 5] "DOS Extra", Issue 4/1993, DMV-Verlag
- [ 6] "Das Aufsteigerbuch" (C64 -> Amiga), Michael Strauch, Alexander  
Stellmach, © 1987 by DATA BECKER GmbH, Düsseldorf.  
ISBN 3-89011-134-4
- [ 7] "Formats.doc" of the ShowVIC distribution on SaarAG-Disk #616.  
ShowVIC is (C)opyright 1993 by Matt Francis.
- [ 8] "Einführung in die digitale Bildverarbeitung", Wolfgang Abmayr,  
© 1994 B.G. Teubner, Stuttgart.  
ISBN 3-519-06138-4
- [ 9] "Noch mehr Dateiformate", Günter Born,  
© 1995 Addison-Wesley (Deutschland) GmbH.  
ISBN 3-89319-757-5
- [10] The Independent JPEG Group's software package(s) with Source-Code(s)  
and Documentation. Release 4 through 6.
- [11] "The Programmer's PC Sourcebook", Thom Hogan,  
published by Microsoft Press, © 1991 by Thom Hogan.  
ISBN 1-55615-321-X
- [12] "C/C++ Users Journal (tm)" (several issues),  
© Miller Freeman Inc.
- [13] ... and perhaps books/magazines/articles, which I don't remember yet !
- [14] ... as well as texts found on AmiNet, BBS or CD-ROM.
- [15] Newsgroups in Fido (mostly \*.GER) and UseNet (mostly DE.\* and Z-Netz)

## 1.87 Credits

Credits:

~~~~~

24BitToHAM.svoperator

This SVOperator bases on code, which has been included with
FBM Release 1.0 25-Feb-90 by Michael Mauldin.
The original code had been written by Harald C. Koch to convert
24 Bit RGB data (FBM format) to HAM6-ILBM files.
I modified it to create format-independent 8 Bit chunky Pixel buffers,
which can be handled by superview-library.

Also, the code has been rewritten to also be abled to create palette-
based HAM8-output.

The code used for the "quick" option of the SVOperator (without palette) is not related in any way to code of the FBM package.

Here's the copyright notice as found in "fbham.c"
(revision headers cut off / left out) :

```
* fbham.c: FBM Release 1.0 25-Feb-90 Michael Mauldin
*
* Copyright (C) 1989,1990 by C. Harald Koch & Michael Mauldin.
* Permission is granted to use this file in whole or in part for
* any purpose, educational, recreational or commercial, provided
* that this copyright notice is retained unchanged. This software
* is available to all free of charge by anonymous FTP and in the
* UUNET archives.
[...]
```

AmiFIG.svobject

This FIG support module accesses the external program "fig2dev", which for example is available as port of the fig port AmiFIG 1.1 from AmiNet, which is (C) 1996 Andreas Schmidt.

C64.svobject

For getting information about the C64 Koala and Doodle formats I took a look into the source code of ComView 1.0 by Paul Grebenc, which can be found as "C64View" on the SaarAG-Disk #523.

I did not include and use the Source Code as such, but I really learnt a much out of it. The algorithms are perhaps nearly the same, but because I do not use file-to-screen decoding my code is perhaps some 100% faster (different structure, many optimizations).

Dither24Bit.svoperator

This SVOperator bases on code, which has been included with FBM Release 1.0 25-Feb-90 by Michael Mauldin. The original code had to be strongly modified to work with common 24 Bit RGB Pixels instead of R-G-B planes and has been made fully re-entrant (also various other changes had to be done). Only the basic algorithm is similar to the original code. Here's the copyright notice as found in "fbquant.c" (revision headers cut off / left out) :

```
/******
* fbquant.c: FBM Release 1.0 25-Feb-90 Michael Mauldin
*
* Copyright (C) 1989,1990 by Michael Mauldin. Permission is granted
```

```

* to use this file in whole or in part for any purpose, educational,
* recreational or commercial, provided that this copyright notice
* is retained unchanged. This software is available to all free of
* charge by anonymous FTP and in the UUNET archives.
[...]
```

EPS.svobject

EPS.svobject uses strongly modified code from

```

/* pnmtops.c - read a portable anymap and produce a PostScript file
**
** Copyright (C) 1989 by Jef Poskanzer.
**
**
[...]
```

which is allowed by its copyright statements.

FBM.svobject

For getting information on the FBM-Format I took a look into the file "fbm.h" and other source-files, which describe this file format and are part of FBM Release 1.0 25-Feb-90 by Michael Mauldin. No source-code from this package - only the "pure information" - has been used for FBM.svobject.

GIF.svobject

For the GIF LWZ Decoding Routines I used some code of the FBM Package. The code of the original routines has been strongly modified and enhanced/improved (there are almost no similarities to the original code left now).

Here's the Copyright notice as found in the file "flgifr.c" (revision headers cut off / left out) :

```

* flgifr.c: FBM Release 1.0 25-Feb-90 Michael Mauldin
*
* Modifications to GIFTORLE are Copyright (C) 1989,1990 by Michael
* Mauldin. Permission is granted to use this file in whole or in
* part for any purpose, educational, recreational or commercial,
* provided that this copyright notice is retained unchanged.
* This software is available to all free of charge by anonymous
* FTP and in the UUNET archives.
*
* Derived from 'giftorle', written by David Koblas
*
* +-----+
* | Copyright 1989, David Koblas. |
* | You may copy this file in whole or in part as long as you |
* | don't try to make money off it, or pretend that you wrote it. |
* +-----+
```


For the GIF LWZ Encoding Routines I used some code of the FBM Package. The code of the original routines has been strongly modified and enhanced/improved.

Here are the Copyright notice of these modules as found in the files "flgife.c" and "flgifc.c" (revision headers cut off / left out) :

```
* flgife.c: FBM Release 1.0 25-Feb-90 Michael Mauldin
*
* Modifications to GIFENCODE are Copyright (C) 1989,1990 by
* Michael Mauldin. Permission is granted to use this file in whole
* or in part for any purpose, educational, recreational or commercial,
* provided that this copyright notice is retained unchanged.
* This software is available to all free of charge by anonymous
* FTP and in the UUNET archives.
*
* flgifc.c: FBM Release 1.0 25-Feb-90 Michael Mauldin
*
* Modifications to GIFENCODE are Copyright (C) 1989,1990 by
* Michael Mauldin. Permission is granted to use this file in whole
* or in part for any purpose, educational, recreational or commercial,
* provided that this copyright notice is retained unchanged.
* This software is available to all free of charge by anonymous
* FTP and in the UUNET archives.
*
* Based on: compress.c - File compression ala IEEE Computer, June 1984.
*
* Spencer W. Thomas      (decvax!harpo!utah-cs!utah-gr!thomas)
* Jim McKie              (decvax!mcvax!jim)
* Steve Davies           (decvax!vax135!petsd!peora!srd)
* Ken Turkowski          (decvax!decwrl!turtlevax!ken)
* James A. Woods         (decvax!ihnp4!ames!jaw)
* Joe Orost              (decvax!vax135!petsd!joe)
*
```

GhostScript.svobject

GhostScript.svobject accesses an external Ghostscript port, like for example Ghostscript 3.53 from AmiNet, which is based on Aladdin Ghostscript. Aladdin Ghostscript is Copyright (C) 1989, 1995 Aladdin Enterprises. All rights reserved.

JPEG.svobject

This software is based in part on the work of the Independent JPEG Group. Release 6 was used as found on AmiNet, former attempts based on V4.

PICT.svobject

The PICT loader module is based on code derived from the PBM package, namely the standalone-module "picttoppm.c". It is said it can be used freely, so I decided to do this instead of trying to implement this weird PICT-2 stuff of QuickDraw

(is it a vector format or a bitmap format ;-)

The original source has been strongly modified to fulfil the needs of being integrated as a library module for superview.library. To be more independent from the pbm.package (great improvements on code size) I also extracted the standard bdf font from the pbm part (also free).

Here are the Copyright notice of these modules as found in the files "picttoppm.c" and "libpbm5.c" (revision headers cut off / left out) :

```
/*
 * picttoppm.c -- convert a MacIntosh PICT file to PPM format.
 *
 * [...]
 *
 * Copyright 1989,1992,1993 George Phillips
 *
 * Permission to use, copy, modify, and distribute this software and its
 * documentation for any purpose and without fee is hereby granted, provided
 * that the above copyright notice appear in all copies and that both that
 * copyright notice and this permission notice appear in supporting
 * documentation. This software is provided "as is" without express or
 * implied warranty.
 *
 * George Phillips <phillips@cs.ubc.ca>
 * Department of Computer Science
 * University of British Columbia
 *
 * $Id: picttoppm.c,v 1.7 1993/10/26 22:40:31 phillips Exp phillips $
 */
```

Only the default font definition has been taken from this one:

```
/* libpbm5.c - pbm utility library part 5
**
** Font routines.
**
** Support for BDF fonts Copyright 1993 by George Phillips.
**
** Copyright (C) 1991 by Jef Poskanzer.
**
** Permission to use, copy, modify, and distribute this software and its
** documentation for any purpose and without fee is hereby granted, provided
** that the above copyright notice appear in all copies and that both that
** copyright notice and this permission notice appear in supporting
** documentation. This software is provided "as is" without express or
** implied warranty.
*/
```

PNG.svobject

Is based on the png reference library, which allows being used e.g. for freely distributable and commercial programs

libpng

Copyright (c) 1995 Guy Eric Schalnat, Group 42, Inc.

Contributing Authors:

Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

RotateFree.svoperator

The basic algorithm for rotation about any given point with any possible angle had been described in "C/C++ Users Journal", issue August 1995. It was strongly dependend on TIFF input, did only work with 8 Bit input and was not able to adjust the image buffer size to prevent image parts to be cut off. Used only the basic algorithm as a hint how to do hit.

TIFF.svobject

Some parts of this software are based in part on code from the "Aldus Developers Desk" Release 90-06-14 (as found on the Nova Media "Grafik-Collection I CDROM", 1993, Directory "ZIP", File "TIFFRD.ZIP").

Many work had to be done, to get that code working on the Commodore Amiga (or even compileable under SAS/C V6.x).

In detail, I'm making use of strongly modified versions of the lzw (lzwde.c) and CCITT 1D (tiff2.c) decoding routines.

YUVN.svobject and PCD.svobject

For the 24 Bit YUV <-> RGB Conversion Routines I used some code of the NetPBM Package.

The code of the original routines has been strongly modified and enhanced/improved (there are almost no similarities to the original code left now, except the main algorithm).

For PCD also the factors to scale YCbCr to RGB had to be adjusted, since the ones of PCD are slightly different to those of JPEG.

Here are the Copyright notices as found in the specific source files :

```
/* ppmtoyuvsplit.c - convert a portable pixmap into 3 raw files:
** - basename.Y : The Luminance chunk at the size of the Image
** - basename.U : The Chrominance chunk U at 1/4
** - basename.V : The Chrominance chunk V at 1/4
** The subsampled U and V values are made by arithmetic mean.
**
** If CCIR601 is defined, the produced YUV triples are scaled again
** to fit into the smaller range of values for this standard.
**
** by A.Beck
** Internet: Andre_Beck@IRS.Inf.TU-Dresden.de
```

```

**
** Based on ppmtoyuv.c
**
** Permission to use, copy, modify, and distribute this software and its
** documentation for any purpose and without fee is hereby granted, provided
** that the above copyright notice appear in all copies and that both that
** copyright notice and this permission notice appear in supporting
** documentation. This software is provided "as is" without express or
** implied warranty.
*/

/* yuvsplittoppm.c - construct a portable pixmap from 3 raw files:
** - basename.Y : The Luminance chunk at the size of the Image
** - basename.U : The Chrominance chunk U at 1/4
** - basename.V : The Chrominance chunk V at 1/4
** The subsampled U and V values are made by arithmetic mean.
**
** If ccir601 is defined, the produced YUV triples have been scaled again
** to fit into the smaller range of values for this standard.
**
** by Marcel Wijkstra <wijkstra@fwi.uva.nl>
**
** Based on ppmtoyuvsplit.c
**
** Permission to use, copy, modify, and distribute this software and its
** documentation for any purpose and without fee is hereby granted, provided
** that the above copyright notice appear in all copies and that both that
** copyright notice and this permission notice appear in supporting
** documentation. This software is provided "as is" without express or
** implied warranty.
*/

```

HilbertDither256.svoperator

The description of the "fractal Hilbert dithering" method has been found in the "mc magazine, issue 6/94, Franzis-Verlag GmbH". The basic techniques (L-System, 16x16 block error approximation) are perhaps the same, but the source code as such has been rewritten completely in almost any of its parts, so that it has not just been "taken out of there" (any limits have been removed; e.g. the original source was a standalone-program, which only worked with TARGA graphics, which had a width and height divideable by 16, but max. 1280x960 or 2560x1920).

1.88 Requirements for the SuperView.library Package

Generally, you need at least an 68000 Amiga, running with OS ↔
2.04+.

Better performance results require better software/hardware.

More requirements (software, hardware, legal):

\textdegree{} For usage of the IFF-based SVObjects, like ILBM, ACBM, DEEP and ↔
YUVN,

you should take care to have `iffparse.library V37+` on your bootdisk.

`\textdegree{}` For `DataType` support with OS 3.x+ you need `datatypes.library V39` ←
 +
 to be present. If it is not present, `datatype-support` is disabled, but you might not need it, anyway.

`\textdegree{}` For being able to use the 24 Bit `datatype` extension for ←
`CyberGraphX`
 - via `CyberDataType.svobject` - you need `picture.datatype V43` to be installed together with a registered `CyberGraphX` system. If you don't have `CyberGraphX`, but `picture.datatype V43` `CyberDataType.svobject` will not work, but it will badly influence the standard `picture.datatype` support of `superview.library` (with several other applications as well).

`\textdegree{}` `Postscript (TM)` support via
`GhostScript.svobject`
 and `EPS.svobject`
 requires `Ghostscript` being correctly installed and configured. Without working `GhostScript.svobject` and a special flag set, `EPS.svobject` will not be able to export any `Postscript (TM)` data from `EPS(F)` files, but only any possibly included preview images from the header.

`\textdegree{}` `AmiFIG (fig)` support requires both, `Ghostscript` and `AmiFIG.svobject`
 ,
 being correctly installed and configured.

`\textdegree{}` `GIF` and `LZW-TIFF` support requires a general (non-existent patent because of countries' law) or personal (if protected by patent claims in country) license for `LZW`. See `Installation` section for how to apply a given or non-existing patent situation to this software. The ability of this software to handle `LZW` is completely in the hand of the user - so he is responsible for any licensing to be done.

Additionally there are some more requirements, which do not necessarily depend on the OS or the hardware:

`\textdegree{}` Some `SVObjects`, `SVDrivers` or `SVOperators` additionally require different hardware/software configurations, but usually this is stated within their own documentation.

Turbo versions will be installed by the installer-script automatically, if it makes sense (CPU auto-detection - may not work with 68060).

1.89 NotesAndHints

Notes and Hints

SuperView-Library sounds impressive, but...

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 SVOBJECTS/DRIVERS/OPERATORS

1.90 usagenote

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

~~~~~

As with V11.8 superview.library will only be distributed together with programs, whose authors did get a license for usage and distribution. This may be Freeware, Shareware or commercial programs.

For example, my program SuperView, which is SHAREWARE and also currently most appropriate Viewer-Program for the library, always comes together with the latest available Library-Version within a subdirectory. You then just have to install to program itself and also the library.

When using more than one program with superview.library-support, take care only to install the most up-to-date versions and not to overwrite them with older ones.

## 1.91 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, SVDIVERS and SVOOPERATORS request as much memory as they need - which directly depends on the size of the processed graphics - and there's no automatic virtual memory manager or such (I introduced some vmem-support it with V9.9, but upto now there aren't any modules making use of it).

But actually nearly no CHIP memory is needed for most of the actions, so 1-2 MB CHIP mem should suffice for almost all actions (if you don't want to display graphics sized 2048x2048x8 with AGA - 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.

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 SVDivers 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 ...

1.92 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, so the savers for non-viewmode-preserving fileformats just don't accept HAM data as input.

The solution

-----

There are two possible ways :

- a) only convert to FileFormats, which preserve the ScreenMode information (currently there are only IFF-ILBM, IFF-ACBM and SVO).
- b) convert (and maybe dither) them before saving, e.g. do a 24 Bit conversion via "AnyTo24Bit" and then - maybe - dither them down to 2..256 colors via "Dither24Bit"

## 1.93 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

You did not set an available or valid 24 Bit SVOperator, which AGA.svdriver - or as well ECS.svdriver - could use for adjusting the number of colors to less than 256.

The solution

Within AGA.controlpad (respectively ECS.controlpad) you may set "24BITOPERATOR=<operatorname>" and specify a specific operator for e.g. dithering 24 Bit graphics to 256 colors, converting them to grayscales or HAM6/8 etc.

The same trick may take place for ECS.svdriver on systems, which can't display more than 16/32(64EHB) colors, so that specifying the (optional) "8BITOPERATOR=<operatorname>" would allow automatic conversion to less colors.

For AGA-only systems it might also be useful to adjust the output of SVOobjects to AGA needs wherever possible.

For example the JPEG.svobject does allow on-the-fly dithering which may be activated via the switch "COLORDEPTH=8" within JPEG.controlpad. Output then will always be done in 256 Colors (or grayscales) only.

The alternative to always do the dithering by hand via calling SVOoperators from SuperView's GUI also is another possible way.

1.94 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 SVOobjects, which support reading of 24 Bit Files do also support writing of 24 Bit Files.

SVOobjects, which support writing 24 Bit data yet are for example:



|      |           |       |
|------|-----------|-------|
| ILBM | JPEG      | Targa |
| BMP  | SunRaster | YUVN  |
| PCX  | FBM       | PNM   |
| SVG  | SGI       | DEEP  |
| TIFF | QRT       | PNG   |

The solution

-----

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.

## 1.95 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 SVOPERATORS

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)