

SuperView-Library

COLLABORATORS

	<i>TITLE :</i> SuperView-Library	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		September 19, 2022
<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	2
1.4 Distribution	2
1.5 Abilities, Purposes and much more	3
1.6 Installation	3
1.7 Overview of current available SVOjects	4
1.8 Overview of current available SVDivers	5
1.9 Software supporting SuperView.library	6
1.10 Thanks, greetings, credits and the rest	6
1.11 How to contact the author	7
1.12 The future of the SuperView.library	7
1.13 History	8
1.14 ILBM.svobject	14
1.15 ACBM.svobject	17
1.16 Datatypes support	19
1.17 PCX.svobject	20
1.18 SVO.svobject	24
1.19 GIF.svobject	27
1.20 BMP.svobject	31
1.21 FBM.svobject	33
1.22 C64.svobject	35
1.23 IMG.svobject	37
1.24 TIFF.svobject	38
1.25 Targa.svobject	42
1.26 WPG.svobject	44
1.27 SunRaster.svobject	45
1.28 Pictor.svobject	46
1.29 MAC.svobject	47

1.30 JPEG.svobject	48
1.31 ECS.svdriver	48
1.32 AGA.svdriver	50
1.33 EGS.svdriver	53
1.34 Used literature for developing this program	56
1.35 Credits	56
1.36 Requirements for the SuperView.library Package	58

Chapter 1

SuperView-Library

1.1 SuperView Library Documentation

superview.library V8.2

- FREeware -

© 1993-94 by Andreas R. Kleinert. All rights reserved.

This program has been written under OS V3.00 and is therefore fully compatible. It needs OS V2.04+.

Release Date : 10.09.1994

Copyrights

Disclaimer

Distribution

Requirements

How to install it

Short: Purpose and Abilities

Supporting Software

SVObject Descriptions/History

SVDriver Descriptions/History

Possible future enhancements ...

History

Thanks and Greetings

Used literature for developing

Credits

How to contact the author

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

1.2 Copyright

The superview.library and all the distributed files (e.g. the documentation files) are (C)opyright 1993-94 by Andreas R. Kleinert. All rights reserved.

The usage of the "superview.library" for own program projects, no matter if commercial or non-commercial, is allowed without any restrictions.

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

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

The program SuperView-Library in this version is freely distributable. You may copy it, if the copyright notice is left intact and all of its parts are included in the distribution.

! Programmers of freely distributable programs, which make use of
! superview.library, may create their own distributions of superview-
! library for inclusion with their programs, but they must state
! within the program's documentation from where the whole and original
! distribution can be obtained (e.g. from AmiNet).
! Programmers should - but need not - send me an EMail reporting this.

This program must not be included in commercial packages or commercial program collections without my written permission.

The program must not be sold in any way, but it is allowed to take a nominal fee including the costs for copying.

! There's one EXCEPTION : If someone writes a commercial program,
! which uses superview.library, he may include it into his program

! package, just like a programmer of a freely distributable program.
! Programmers should - but need not - send me an EMail reporting this.
! I also would like to get a version of the final program, but this
! is really voluntarily for the authors of the programs.

The program may be put on public domain disks or included in public domain disk libraries.

Special permission hereby goes to Fred Fish's AmigaLib-Disks and the german series (in alphabetical order) :
AmigaSzene, BerndsPD, FRANZ, GPD, SaarAG, TAIFUN and TIME.

This program may also be distributed via electronic mail and may be put into mailboxes as long as the redistribution conditions are respected in all points.

By using or distributing this program you automatically agree to all of the above conditions and terms.

1.5 Abilities, Purposes and much more

The "superview.library" is used in many of my programs and it consists of many functions, which make displaying, saving and converting of various picture formats (IFF-ILBM, IFF-ACBM, PCX, ...) much easier. External Viewer-Libraries (called "svobjects") and Graphic Card Drivers (called "svdrivers") allow highest possible flexibility and easy to do expansions.

It 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.

Just send me your own SVOjects and SVDdrivers if you want to have them added to the main distribution archive or put onto the developer disk.

1.6 Installation

Just use the supplied Installer-Script, or "by hand"-copy "superview.library" and "superviewsupport.library" to your "LIBS:"-Directory, the svobjects to "LIBS:svobjects/" and the svdrivers to "LIBS:svdrivers/" !

"datatypes.library" V39+ is only needed, if OS V3.00+ DataTypes are wished to be supported.

Please take care, which libraries are also additionally needed by the single SVOjects!!!

"AddBuffers"-Values for the used Drives should at least be around 25.

You may set the following environment variables at the location

"ENV:superview-library". SuperViewPrefs can be used for this.

Name	Content
Default.svdriver	Name of default-ScreenDriver. Read by superview-library at first opening. MUST NOT contain any extra characters (spaces, tabs, etc.).

[There are also several SVOjects/SVDrivers, which support environment variables. Default-values will be used, if these are not set.]

1.7 Overview of current available SVOjects

Available SVOjects (as far as known) :

SVOject	Type	Read-Support	Write-Support
IFF-ILBM	EXT	(system-dependent)	(system-dependent)
IFF-ACBM	EXT	(system-dependent)	(system-dependent)
OS V3.00+ DataTypes	INT	all Pictures	(not available)
PCX upto V3.0	EXT	max. 8/24bit Files	max. 8bit Files
SVO upto V1.0	EXT	(system-dependent)+24	(system-dependent)+24
GIF upto 89a	EXT	max 8bit Files	max 8bit Files
BMP (Windows)	EXT	max 8/24bit Files	max 8/24bit Files
FBM (Unix)	EXT	max 8bit Files	max 8bit Files
C64 (Koala, Doodle)	EXT	max 4bit Files	(not yet available)
IMG (GEM/Ventura)	EXT	max 8bit Files	(not yet available)
TIFF (V5.0)	EXT	max 8/24bit Files	max 8bit Files
Targa	EXT	max 8/24bit Files	max 8/24bit Files
WPG (WordPerfect)			

```

EXT    max 8bit Files          (not yet available)

SunRaster (RAS)
EXT    max 8/24bit Files      max 8/24bit Files

Pictor/PCPaint/PIC
EXT    max 8bit Files          (not yet available)

MAC (MacPaint)
EXT    max 1bit Files          (not yet available)

JPEG (JFIF)
EXT    max 8/24bit Files      always as 24bit Files

```

JPEG.svobject is available in a separate archive
(writing possible since V2.3).

Planned SVObjects (no guarantee !) :

- PBM/PGM/PNM/PPM
- RIFF
- MTV
- SGI
- RLE
- and more

1.8 Overview of current available SVDivers

Available SVDivers (as far as known) :

SVDriver	Requirements	BITPLANE	ONEPLANE
ECS	ECS, OS V2.04+ (V37), 68000+	(system)	8bit
AGA	AGA, OS V3.00+ (V39), 68020+	(system)	8/24bit
EGS-System	EGS-Graphic-Libraries	8bit	8/24bit

Planned SVDivers (no guarantee !) :

- Picasso
- Opalvision
- Retina
- and more ;)

1.9 Software supporting SuperView.library

The following programs are currently supporting the "superview.library" :

- SuperView
- SimpleView (superview.library Example Program)
- MicroView (superview.library Example Program)
- KFracPlus
- SIP (for examinations)
- DRAFU (unreleased Beta-Version)
- ...

1.10 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 used to design the GUI of SuperViewPrefs.

* 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 new AmigaGuide Documentation... ;)

* 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 ;-)

* and last NOT LEAST

- all `_registered_` users of SuperView for supporting Shareware !!

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

Alex	Carbin	TuC / Co-Sysop Century
Sven	Drieling	Indy - Creator of "PowerBrei" DiskMag
Fred	Fish	AmigaLibDisks
Stefan	Grad	GPD-Disks
Paul	Huxham	(Co-)Programmer of OpalVision-SVDriver, Australia
Stefan	Kremer	TuC / Sysop Century
Alex	Lange	Time PD-Disks
H.P.	Lattka	Franz PD-Disks
Andreas	Manewaldt	Taifun PD-Disks
Axel	Melzener	Game Object Design
Michael	Petrikowski	Amiga Szene PD-Disks

Thomas Saenger Sysop of "elephant BBS", my Fido-Boss
 Steve Quartly (Co-)Programmer of OpalVision-SVDriver, Australia
 NJ Verenini The one, who created this nice "Spumoni's Workbench"
 JPEG-Picture with SuperView ... :-)

- some users, which reported bugs via mail, e-mail or telephone
 (only the ones, which have not already been listed above) :

Thomas Alexnat
 Rudi Brand
 Aaron Digulla
 Thomas Gundlach
 Richard Hartmann
 Reinhard Theling
 .
 .
 .

- thanks also go to Mira, Steffi, Tanja and Teresa for no special
 reason... :-)

1.11 How to contact the author

If you like, you may send me some money. Perhaps this will motivate me to continue programming such programs or just making updates of this one. Send bug-reports, money or whatever to :

Andreas R. Kleinert,
 Grube Hohe Grethe 23,
 D-57074 Siegen,
 Germany.

E-Mail : Andreas Kleinert (Fido 2:2457/345.10)
 Andreas.Kleinert@p10.f345.n2457.z2.fido.hotb.sub.org
 Andreas_Kleinert@f345.n2457.z2.fido.sub.org
 Andreas_Kleinert@p10.f345.n2457.z2.fidonet.org

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 500/1000/.../4000T, Kickstart/OS Version, RAM, HardDisk, special Hardware/Software).

1.12 The future of the SuperView.library

Future enhancements :
 =====

All parts will be made bigger, better, faster and more!

(Of course you're invited to help doing so ... ;-))

1.13 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	-

V8.2 (10.09.1994) :

- rewrote SVObject-/SVDriver-Ref-Docs a little bit :
these have been so strange and weird in some parts,
that it's been really necessary (still not perfect)
- there's a bad bug in the OS-Routines for handling chunky pixel
graphics: when passing a width, which does not end on a
Byte-Boundary, the graphic will be trashed !
I tried all possible ways to fix this, but it did not work.
Now the last upto 7 Pixels of such graphics simply will be
cut off until I find a solution (that's perhaps better than
always getting garbage on the Screen, which then will be followed
by a "memory" guru).
This concerns all Intuition-SVDrivers (ECS, AGA) and all
ONEPLANE-Routines of superviewsupport.library (which are e.g. used
by SVObjects for converting GfxBuffers before saving).
--> Thanks to Rudi Brand, which sent me this special GIF picture,
which enabled me to find this bug, finally (searched for
the reason a long time).
- found a bug, which I could not reproduce yet: Aaron Digulla
wrote, that superview.library always crashes, when there's
more than one SVDriver installed.
Are there more people, which are concerned to this bug ?
- TOTALLY : - bug-fixed/improved SVDrivers : ECS, AGA
 - bug-fixed/improved SVObjects : PCX

V8.1 (13.08.1994) :

- "thanks & greetings" list has been become bigger ;-)
- made some "cosmetic changes" to this GuideFile ...
- slightly changed internal file recognition-procedure:
modified internal SVLI_GetFileType() routine a little
and exported it as "SVL_GetFileType()".
This now does allow easy decisions, whether a file might
be handled by superview.library or not.
SVL_GetFileType() writes the filetype (NOT subtype) into
a variable and might return SVERR-Codes (e.g. "file not found").
--> Sorry, Steve : Changed concept a little bit (not only
true/false now), but should still suffice !?

V7.2 (22.07.1994) :

- improved/bug-fixed SVOjects : GIF, ...
(bad bug in writing GIFs. Only occurred when writing more than one or two in a single session).

V7.1 (20.07.1994) :

- added four new functions for adding and removing of SVOjects/SVDrivers
- fixed medium bug in "SVL_GetSVObjectList()" function (incomplete list, small memory loss)
- fixed some minor bugs in the docs and elsewhere
- totally re-wrote SuperViewPrefs (see documentation there)
- added AmigaGuide documentation, which has been created by Gerd Frank (thanks for your work, Gerd !)
I only did some small changes, which seemed important to me.
- improved/bug-fixed SVOjects : GIF, ...
- improved/bug-fixed SVDrivers : EGS, ...
- re-worked all Example-Programs
- fixed small, but strange bug (?) in file-recognition routine :
when facing an unknown file-type, some information may have been read from location zero (for switch-statement), but never caused any problems (luckily !).
This "seems-to-have-never-appeared-but-fixed-nevertheless"-bug was perhaps a result of the V4.1-bugfix.

V6.5 (04.07.1994) :

- includes superviewsupport.library V4.1
- moved "reference"-docfiles to "Programmers/Docs"-directory
- added more detailed description of SV_GfxBuffer structure to <superview/superview.h>
- added ".fd"-files to "Programmers/fd", which might be useful for Basic programmers (".bmap" generation) or for generating Link-Libraries
- Env-Variable is "Default.svdriver", not "SVDriver.ascii" !!
- included SunRaster.svobject (RAS). Already with full write-support.
- improved/bug-fixed SVOjects : SVO, Targa, ...
- improved/bug-fixed SVDrivers : AGA, ...
- last not least : I'm now reachable via EMail
(Fido : 2:2457/245.10)

V6.4 (13.06.1994) :

- revised Example-Programs
 - AGA.svdriver now at least needs 68020+
(there are no AGA machines, which do not have one ...)
 - strongly improved and enhanced SVOjects : BMP, Targa
 - attention : new JPEG.svobject (V2.3) now also is able to write JPEGs !
-

- included Pictor.svobject (PCPaint). Read only yet.

V6.3 (06.06.1994) :

- small bug-fixes
- in all of the supplied SVOjects with write-support there are now also the flags for indicating GfxBuffer-Support set

V6.2 (03.06.1994) :

- improved GIF.svobject in some ways
- revised TIFF.svobject and fixed some bad bugs (also added some new features, which have not been largely tested yet)
- fixed bad bug in ILBM.svobject (concerned saving of odd-width Pictures). There's perhaps the same bug in ACBM.svobject, but why making it incompatible with older versions ? There might be more of these "word-padding bugs" in other SVOjects, but not in all cases the specifications of the file formats are as clear about this as this is the case with IFF-ILBM ...
- added FAQ-like documentation (NotesAndHints.guide)
- removed example-program "ReadGfxBuffer", because since V6.1 this is also included in "SimpleView"'s source-code

V6.1 (28.05.1994) :

- FIXED BAD BUG : SVL_SetGfxBuffer() plus SVL_SuperWrite() FAILED always, which was (again) caused by a very simple bug, not worth to mention
 - with all of the supplied SVOjects with write-support it is now possible to save not only Screens, but also GfxBuffers : all their internal write routines have been completely rewritten, so that supplied Screens are transferred to GfxBuffers before saving. BitPlane AND OnePlane Buffers upto 8 Bit are supported : they are converted to each other by using the appropriate functions of superviewsupport.library V3+.
 - this great enhancement - I was working on it since the introduction of superview.library V3+ - has been the reason, why the version number increased to 6 !
You may use this feature with the supplied version of SimpleView or with SuperView V2.5+ !
 - added four new functions for simple allocation and delocation of simplified forms of SVOject and SVDriver lists.
This will make accessing of superview-library's write-functions much more easier for custom applications.
 - added full (reference-like) documentation for SVOjects and SVDrivers as such
 - a JPEG.svobject is available since V5.2 (separately)
 - and more ...
-

V5.2 (22.05.1994) :

- superviewsupport.library now contains four new functions to allow support of the new ControlPad-FileFormat for SVOBJECT/SVDRIVER/... Configuration Files. (See Docs/SVSupport-Docs)
These Files are almost similar to Tooltype-Lists.
- removed one totally WRONG paragraph from Reference_ENG.doc, which concerned the SVL_InitHandleAsClip() function (the remark still referred to the unreleased beta version 3.9, which was incompatible)
- included Targa.svobject with 24 Bit-Support. Read only yet.
- bug-fixed and enhanced 24 Bit-Support of PCX.svobject.
- improved AGA.svdriver slightly (nearly a bug-fix)

V5.1 (18.05.1994) :

- FIXED BAD BUG : SVL_ReadGfxBuffer() failed ALWAYS, which was caused by a very simple bug, not worth to mention (Murphys Law ?)
- there was also a bug in checking the SVOBJECTS's versions for using the GfxBuffer-Functions : we checked for V3+ instead of V2+
- fixed bug in ACBM.svobject
- fixed some Doc-Files

V4.1 (15.05.1994) :

- because of changing/extending specifications : Jumping to V4
- included MAC.svobject. Read-Only yet.
- all NON-DISK loading and saving, e.g. from/into Clipboard DID NOT WORK ANY LONGER since V2.4.
Before ILBM.svobject and ACBM.svobject had been externalized, this feature worked, but only with these.
After externalization (with this version) this bug was detected and fixed.
Specifications had to be changed slightly - in a compatible way - to re-arrange READING to work again (see Reference_ENG.doc, SV_InitHandleAsClip-Function).
For WRITING only superview-Library had to be changed.

V3.8 (12.05.1994) :

- use of SAS/C V6.51
 - created new "superviewsupport.library", which is used by superview.library and all supplied SVOBJECTS and SVDRIVERS.
This saves between 500 and 1600 Bytes for each svobject and also results in 1000 Bytes for superview.library.
superviewsupport.library itself only needs about 3000 Bytes, so there's a great gain left.
-

V3.7 (01.05.1994) :

- this version includes all "last minute bug-fixes" to V3.6 and more
- improved some SVOjects a lot : PCX, TIFF, ...
- improved some SVDdrivers a lot : EGS
- slightly improved SuperViewPrefs
- changed Installer-Script to create "ENVARC:Default.svdriver", for people who forget to use SuperViewPrefs

V3.6 (18.04.1994) :

- this version includes all "last minute bug-fixes" to V3.5 and more
- included TIFF.svobject. Read-Only yet.
(first version already included as bug-fix to V3.5)
- included WPG.svobject. Read-Only yet.
- improved some SVOjects a lot : IMG, TIFF, GIF, ...
- improved some SVDdrivers a lot : AGA, EGS
- added 24 Bit-Support (ONEPLANE-RGB) to AGA- and EGS-SVDriver

V3.5 (27.03.1994) :

- new Example-Program : MicroView !
- Datatypes-Support now just for PICTURES : everything else will be rejected
- updated Include-Files to reflect V3.5, made some changes
- all supplied SVOjects do no longer have a default SVDriver integrated : this saves about 1600 Bytes disk space, which we need for future expansions
- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one. This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit, but increases speed on 68020+ systems
(should also decrease codesize, but does not ...)

V3.4 (25.03.1994) :

- this version includes all "last minute bug-fixes" to V3.3 and more
- included IMG.svobject (GEM). Read-Only yet.
- added write-support to FBM.svobject (Unix).
- made MANY changes to ALL SVDdrivers and SVOjects

V3.3 (22.03.1994) :

- improved Example-Sources (SimpleView, ListSVOs, ListSVDs) :
 - no longer dependent on "ak_gen0.library", removed dead code
 - improved internal printf-replacement routine
 - included FBM.svobject (Unix). Read-Only yet.
-

- fixed LibID : superview.library was actually not created in 2994,
but in 1994 ;)

V3.2 (17.03.1994) :

- this version includes all "last minute bug-fixes" to V3.1 and more
- added EGS.svdriver to distribution (very first beginning for support of the EGS graphic system, 8-Bit only yet)
- included C64.svobject (Koala, Doodle). Read-Only. That's NOT a joke !

V3.1 (11.03.1994) :

- Beta-Versions : V2.7
(Parameter-BugFix in SVO_SetWriteSubType() :
pragmas put it into D1 instead of A2)
- integration of the SVDivers
- fixed small bug in memory management routines
- several important fixes and improvements inside the Version2-SVObjects
- bug-fix in superview.library :
a call to SVL_WindowIDCMP, SVL_WindowFlags or SVL_ScreenType resulted in default-values, if an external SVObject was used for this (forget to set these values inside SVL_SuperView again for the SVObject)
- V39-DataType-Displaying now with correct Colors
- more and detailed information of V39-DataTypes
- all currently included SVObjects (except DataTypes) now allow Reading (to Screen or Buffer) and Writing (from Screen) of Graphics

V2.6 (13.02.1994) :

- small enhancements

V2.5 (24.01.1994) :

- included BMP.svobject (read-only)
- little improvements

V2.4 (22.01.1994) :

- recompilation : use of SAS/C V6.5
 - again : fixed bad bug in SVL_FreeHandle(),
external SVObjects would never be closed
 - "externalized" IFF-ILBM and IFF-ACBM as ILBM.svobject and ACBM.svobject
-

V2.3 (16.01.1994) :

- improved speed with writing IFF-ILBM and IFF-ACBM
- fixed bug inside SVL_FreeHandle() : SVO_FreeHandle() was not called, but SVO_CloseDisplay() and SVO_FreeResources()
- fixed bug in Includes (was APTR * instead of APTR sometimes)

V2.2 :

- included GIF.svobject (upto GIF89a, read-only)
- improved speed with displaying IFF-ILBM and IFF-ACBM

V2.1 :

- Beta-Versions : V1.2, V1.3
- integration of the svobjects
- Library may be used by anybody for anything
- complete internal revision
- added DataType-Support (read-only)
- fixed bug in ScreenMode Validation Routine for IFF-ILBM/-ACBM
- new "Best ScreenMode"-Routine for IFF-ILBM/-ACBM und PCX.svobject (different behaviour under OS V37+ and OS V39+)
- included PCX.svobject
- included SVO.svobject plus SourceCode

V1.1 :

- First Version. FREeware.

1.14 ILBM.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREeware. All rights reserved.

Version : 2.11
Release Date : 12.06.1994

Description

~~~~~

ILBM.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and

---

needs any SVDriver with Bitplane-Support.

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

Reading :

- IFF-ILBM uncompressed
- IFF-ILBM CmpByteRun1 compressed

(except 24 Bit-ILBMs)

Writing :

- IFF-ILBM uncompressed
- IFF-ILBM CmpByteRun1 compressed

(except 24 Bit-ILBMs)

History

~~~~~

V2.11 (12.06.1994) :

- now GfxBuffer-writing no longer results in wrong x/yAspect values

V2.10 (05.06.1994) :

- SVOobject flags now reflect, that GfxBuffers can be written
- now "aspect"-values are set to non-zero, if writing an ILBM from a GfxBuffer

V2.9 (03.06.1994) :

- files with odd width would have been saved without pad-bytes : this caused no problems when reading them into ILBM.svobject, but almost all other program might have complained about a "corrupt IFF file". Files now are written padded to word boundaries. Even CBM's "MultiView" and "ILBMLoad" now seem to be pleased with the "odd-width" files written by this program.
- reading of uncompressed padded "odd-width" files would not have worked, now it works, but un-padded files will no longer be read correctly. Compressed files are ALWAYS read correctly, no matter, if they are padded or not (because I do not use the standard decoding routines :-)

V2.8 (27.05.1994) :

- now saving of GfxBuffers is possible : internal write routines have been completely rewritten, so that supplied Screens are transferred to GfxBuffers before saving. BitPlane AND OnePlane Buffers upto 8 Bit are supported : they are converted to each other by using the appropriate functions of superviewsupport.library V3+.
-

V2.7 (15.05.1994) :

- the ClipBoard feature did not work, since ILBM.svobject and ACBM.svobject had been externalized. Specifications had to be changed slightly - in a compatible way - to re-arrange READING to work again. For WRITING only superview-Library had to be changed.
- Besides : the ClipBoard would not have been closed correctly, anyway, because - yet another bug - there was a function call missing.

V2.6 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1
This saves some space again (here : ca. 1100 Bytes).

V2.5 (02.04.1994) :

- improved speed of CmpByteRun1 decoding routines
- improved adjusting of unencoded picture data
- un"#define"d CRNG support routines, which are not used since V2.1 anyway. This saves some code again.
- un"#define"d some more (yet) dead code (AUTH, ANNO, ...)

V2.4 (27.03.1994) :

- removed internal SVDriver-Replacement, so that we now are really dependent on external SVDrivers in libs:svdrivers. This saves about 1600 Bytes disk space and reduces redundancy.
- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one. This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit, but increases speed on 68020+ systems (should also decrease codesize, but does not ...)

V2.3 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine

V2.2 (17.03.1994) :

- fixed "large memory usage" bug in internal ECS-ChunkyPixel routine
- added several security checks (zero-pointers)

V2.1 (09.03.1994) :

- now with SVDriver-Support. Not yet necessarily needed.
 - Clipboard-Support was deactivated accidentally. Now is no longer.
-

- fixed small bug in memory management routines
- fixed bug in reading and writing "big width" pictures
(> Standard-Width for the specific ScreenMode) :
BitMap->BytesPerRow was bigger than thought (for 736 was 96, not 92,
so e.g. saving will result in an Width of 768)
- now sets BMHDF_CMAPOK in bmhd->flags (former pad1) if running under V39
and writing 32 Bit-ColorMaps
- removed "ColorMap replacement" routine
- internal routines : AUTOSCROLL now really works with big-sized Screens
- ColorCycling (CRNG)-Support removed
- fixed several other little bugs

V1.2 (06.02.1994) :

- File Information did not work

V1.1 (16.01.1994) :

- first version

1.15 ACBM.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 2.11
Release Date : 12.06.1994

Description

~~~~~

ACBM.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and  
needs any SVDriver with Bitplane-Support.

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

Reading :

- IFF-ACBM uncompressed  
  
(except 24 Bit-ACBMs)

Writing :

- IFF-ACBM uncompressed  
  
(except 24 Bit-ACBMs)

## History

~~~~~

V2.11 (12.06.1994) :

- now GfxBuffer-writing no longer results in wrong x/yAspect values

V2.10 (05.06.1994) :

- SVOBJECT flags now reflect, that GfxBuffers can be written
- now "aspect"-values are set to non-zero, if writing an ACBM from a GfxBuffer

V2.9 (27.05.1994) :

- now saving of GfxBuffers is possible : internal write routines have been completely rewritten, so that supplied Screens are transferred to GfxBuffers before saving. BitPlane AND OnePlane Buffers upto 8 Bit are supported : they are converted to each other by using the appropriate functions of superviewsupport.library V3+.
- this should also fix the problems with strange-sized pictures
- version still reflected 2.7 instead 2.8. Now we're at 2.9 :-)

V2.8 (15.05.1994) :

- AARGGHH : Reading of ACBM-Files did not work, because ULONG-array checked for wrong IFF-IDs (ID_ILBM) !

V2.7 (15.05.1994) :

- the ClipBoard feature did not work, since ILBM.svobject and ACBM.svobject had been externalized. Specifications had to be changed slightly - in a compatible way - to re-arrange READING to work again. For WRITING only superview-Library had to be changed.
- Besides : the ClipBoard would not have been closed correctly, anyway, because - yet another bug - there was a function call missing.

V2.6 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1
This saves some space again (here : ca. 1100 Bytes).

V2.5 (02.04.1994) :

- un"#define"d CRNG support routines, which are not used since V2.1 anyway. This saves some code again.
 - un"#define"d some more (yet) dead code (AUTH, ANNO, ...)
-

V2.4 (27.03.1994) :

- removed internal SVDriver-Replacement, so that we now are really dependent on external SVDrivers in libs:svdrivers. This saves about 1600 Bytes disk space and reduces redundancy.
- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one. This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit, but increases speed on 68020+ systems (should also decrease codesize, but does not ...)

V2.3 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine

V2.2 (17.03.1994) :

- fixed "large memory usage" bug in internal ECS-ChunkyPixel routine
- added several security checks (zero-pointers)

V2.1 (09.03.1994) :

- now with SVDriver-Support. Not yet necessarily needed.
- Clipboard-Support was deactivated accidentally. Now is no longer.
- fixed small bug in memory management routines
- fixed bug in reading and writing "big width" pictures (> Standard-Width for the specific ScreenMode) : BitMap->BytesPerRow was bigger than thought (for 736 was 96, not 92, so e.g. saving will result in an Width of 768)
- now sets BMHDF_CMAPOK in bmhd->flags (former pad1) if running under V39 and writing 32 Bit-ColorMaps
- removed "ColorMap replacement" routine
- internal routines : AUTOSCROLL now really works with big-sized Screens
- ColorCycling (CRNG)-Support removed
- fixed several other little bugs

V1.2 (06.02.1994) :

- File Information did not work

V1.1 (16.01.1994) :

- first version

1.16 Datatypes support

The program optional uses the OS3.00+ Datatypes. But only the Datatypes for pictures are supported. This means, no Samples, Text, Amigaguide File or anything else will be displayed via the SuperView.Library!

1.17 PCX.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEMWARE. All rights reserved.

Version : 2.14
Release Date : 09.09.1994

Description

~~~~~

PCX.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with Bitplane-/ChunkyPixel-Support.

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

#### Reading :

- 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 - no matter, which WriteSubType has been specified - 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                |

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.

### Remarks

~~~~~

- Why writing only 16/256 Color files ?

When loading 4 Color PCX files, some programs might expect a CGA style color palette, while we only write EGA/VGA palettes. For 8 Color graphics it is not clear how they should be handled.

Also, most programs do not support 32 .. 128 Color graphics, perhaps because it is not clear, how the colormap has to be stored : all at the end of the file or one hunk in the header and only the rest at the end of the file ?

So why should PCX.svobject support things, which are not defined as clear as they should be ?

- Then, why reading 4/8 Color files nevertheless ?

Because PCX.svobject wrote them - this was my fault - before I fixed this bug. So it still reads them, but no longer writes them.

- What about PCC-Brush-Files ?

These should also be supported, because they're just a variation of PCX-Files - usually only the extension is changed - but it seems to me, that some of them are not correctly written, so that reading may sometimes result in garbage.

- What about 24 Bit-Files :

Reading of RLE-encoded files works. It has been tested.
Reading of uncompressed files should now works, due to the bug-fix in V2.11, although it has not been tested yet.
Send me your Public Domain 24 Bit Pictures, if they do not work with PCX.svobject yet !

History

~~~~~

V2.14 (09.09.1994) :

- fixed bug in reading one of the many PCX derivates : files, which contain RLE-codes which compress across lines (some planar files with colors <= 16) now will be read correctly (no longer garbage on the screen or "memory corrupt"). Algorithm had to be changed slightly to allow universal handling, but should not be slowed down significantly (because also some optimizations have been done ;-)
- also some PCC-brushes might now be read (better) ?! (don't know whether those, which work now also did work before ... ;-))

V2.13 (05.06.1994) :

- SVObject flags now reflect, that GfxBuffers can be written

V2.12 (27.05.1994) :

- now saving of GfxBuffers is possible : internal write routines have been completely rewritten, so that supplied Screens are transferred to GfxBuffers before saving. BitPlane AND OnePlane Buffers upto 8 Bit are supported : they are converted to each other by using the

appropriate functions of superviewsupport.library V3+.

- if RLE-encoding is ineffective (output data nearly as large or even larger than input data), the files will be written unencoded now

V2.11 (22.05.1994) :

- added code for reading of RLE-encoded 24 Bit Files (I tested it : it works)
- (hopefully) fixed bug in code for reading uncompressed 24 Bit Files (it forgot to adjust the code to the way, how 24 Bit Data is stored in the PCX FileFormat)

V2.10 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1  
This saves some space again (here : ca. 1600 Bytes).

V2.9 (01.05.1994) :

- slightly changed writing and RLE encoding routines
- added note about PCC files to documentation

V2.8 (23.04.1994) :

- created new hierarchy for decoding routines : they are now much more flexible and error-resistant. Future expansions will be easier to make.
- temporary buffer for rawdata is now delocated immediately after use
- simplified ColorMap reading/handling (and GreyScale generation)
- added support for 24 Bit graphics. Not yet tested.
- Fixed FileInfoRequest() :
  - PCX V2.8b was reported to be "V2.5 without Palette". Fixed.
  - "ColorPlanes" was reported to be "Colordepth". Fixed.  
Now there are four different values for all : pixelbits, pixelbits-sized planes, colordepth and colornum.  
So the user may reconstruct the original file structure in mind.

V2.7 (18.04.1994) :

- changed internal creation/handling of ColorMaps :  
now PCX 2.8b with Black&White or GreyScaled BitMap-Data is really supported.  
Side effect : simplified handling of different PixelBits/Plane combinations

V2.6 (02.04.1994) :

- improved speed of RLE decoding routines
-

V2.5 (27.03.1994) :

- removed internal SVDriver-Replacement, so that we now are really dependent on external SVDrivers in libs:svdrivers. This saves about 1600 Bytes disk space and reduces redundancy.
- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one. This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit, but increases speed on 68020+ systems (should also decrease codesize, but does not ...)

V2.4 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine

V2.3 (18.03.1994) :

- removed some dead code (not compiled nor linked ?!)

V2.2 (17.03.1994) :

- fixed "large memory usage" bug in internal ECS-ChunkyPixel routine
- added several security checks (zero-pointers)

V2.1 (09.03.1994) :

- now with SVDriver-Support. Not yet necessarily needed.
- improved internal "Chunky Pixel to BitMap" (8 Bit) Routine
- writing routine(s) :
  - added code for writing 256 Color graphics (chunky pixel)
  - 2, 4, 8, 32, 64 or 128 Colors are expanded to either 16 or 256 Colors before writing
  - 16 Colors now written as V2.8a, also fixed some small bugs
  - increased speed for writing 16 Colors
  - fixed bug in setting map for 16 Colors (same as in reading routine, which was fixed in V1.2)
- fixed small bug in memory management routines
- internal routines : AUTOSCROLL now really works with big-sized Screens
- fixed several little bugs

V1.3 (24.01.1993) :

- fixed bug with setting ColorMap under V37/V38

V1.2 (02.01.1993) :

- fixed "wrong colors" bug
  - improved speed
-

V1.1 (19.12.1993) :

- first version

## 1.18 SVO.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEMWARE. All rights reserved.

Version : 2.8  
Release Date : 01.07.1994

### Description

~~~~~

SVO.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with Bitplane-/ChunkyPixel-Support.

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

Reading :

- SVO V1.0 ONEPLANE/BITPLANE (includes 24 Bit Files)

Writing :

- SVO V1.0 ONEPLANE/BITPLANE (includes 24 Bit Files)

ControlPad-Switches

~~~~~

ControlPad-Name : "ENV:superview-library/SVO.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

~~~~~

Pictures in the "SVO Graphics File Format" consist of two Files :
The Header-File (<Picture>.svo) and the Data-File (<Picture>).
The Header-File contains information about the Raw-Data in the
Data-File. For loading and saving the Graphics it makes no difference,
which FileName - Header-File or Data-File - is given.
This is managed by the svobject.

Because recognition of the Types of packed Files is only possible by reading the whole File into a buffer and then check it, we use two different Files for both : Graphics Information and Graphics. The Data-File CAN be packed, the Header-File MUST NOT be packed.

The SVO.object does NOT save packed data by default, but you may either specify an appropriate Xpk-Packer in the ControlPad or just pack it with a Xpk-Packer-Program after writing.

The Data saved into the <Picture> File is read from a SV_GfxBuffer structure, which has either been supplied by the calling application or has been created from a supplied Screen-Pointer.

This data may be of type BITPLANE (like in an IFF-ACBM : BitPlane after BitPlane, but without padding) or ONEPLANE (8 Bit ChunkyPixel or 24 Bit RGB-Data).

This allows efficient packing and unpacking with PowerPacker and all other XPK-Packers.

Remember, that the original Data written by the SVO.object may not necessarily be packed and therefore might need a lot of disk space. So better pack it immediately after it has been written.

Source-Code

~~~~~

The "SVO Graphics File Format" is an "Open File Format", which means, that everyone is invented to use and support it and to make suggestions to improve the File Format or the Algorithms to handle it.

To allow support of the File Format and to give an example, how to write your own superview.library compatible "svobjects", this distribution contains the full SourceCode of the "SVO.svobject".

Feel free to use it for your own, commercial or non-commercial, programs !

#### History

~~~~~

V2.8 (01.07.1994) :

- writing of SVO-Files did not work since V2.6
- now it may be specified in "ENV:superview-Library/SVO.controlpad", whether written SVO-Files should be packed and which packer should be used ("PACKMETHOD=xxxx")
- SVO-Files now are written in the way the data has been stored in the GfxBuffer (either ONEPLANE or BITPLANE). Screens are transferred to BITPLANE data. Reading of such files was already possible, but not yet writing. This will sometimes allow more efficient packing with XPK (e.g. when ONEPLANE-8 data with only 16 Colors is used as input).

V2.7 (05.06.1994) :

- SVOobject flags now reflect, that GfxBuffers can be written

V2.6 (27.05.1994) :

- 17.05.94 : modified Doc-File slightly
- now saving of GfxBuffers is possible : internal write routines have been completely rewritten, so that supplied Screens are transferred to GfxBuffers before saving. BitPlane AND OnePlane Buffers upto 8 Bit are supported : they are converted to each other by using the appropriate functions of superviewsupport.library V3+.

V2.5 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1
This saves some space again (here : ca. 500 Bytes).
- version still reflected 2.3 instead 2.4. Now we're at 2.5 :-)

V2.4 (27.03.1994) :

- removed internal SVDriver-Replacement, so that we now are really dependent on external SVDrivers in libs:svdrivers.
This saves about 1600 Bytes disk space and reduces redundancy.
- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one.
This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit, but increases speed on 68020+ systems
(should also decrease codesize, but does not ...)

V2.3 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine

V2.2 (17.03.1994) :

- fixed "large memory usage" bug in internal ECS-ChunkyPixel routine
- added several security checks (zero-pointers)

V2.1 (09.03.1994) :

- very first "version 2" SVOobject with SVDriver-Support
- found bug in the SVO-FileFormat, when storing the 16-Bit and 32-Bit ModeIDs :
both are just defined as 16-Bit-UWORDS, so that the 32-Bit ModeID is theoretically `_scratch_` !
It does seem to work with SAS/C V6.5 nevertheless, but with this version the following work-around is done :
The former "svo_Pad1" field becomes "svo_Version", which indicates extensions to the old 822-Byte-Header.
With V1 a new 32-Byte-ModeID field is added, so that the old one

becomes obsolete for reading of newer files.

The OLD ViewMode32-Field is still written as before, but will only be read when handling files with version < 1 (= (BYTE)0).

- internal routines : AUTOSCROLL now really works with big-sized Screens
- fixed small bug in memory management routines

V1.4 (24.01.1994) :

- little internal improvements

V1.3 (16.01.1994) :

- improved speed

V1.2 (04.01.1994) :

- changed to recognize (reject) possible enhanced, future SVO Files (16-24 Bit)

V1.1 (19.12.1993) :

- first version

1.19 GIF.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany
(Also see notes under "Credits".)

FREEWARE. All rights reserved.

Version : 2.12
Release Date : 22.07.1994

Description

~~~~~

GIF.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with ChunkyPixel-Support.

It supports reading and writing of GIF Files.  
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           |

A special 68030+ Version of GIF.svobject is called "GIF.svobject.030" and has to be copied/renamed to GIF.svobject before usage (it runs on the 68020 as well, but might be more suitable for the 68030 : Only the Compiler knows about it ... ;).

GIF.svobject works fine with Instruction Cache (Burst) and Data Cache (Burst).

Bugs and other Problems

~~~~~

Fixed : The Decoding Routines are fully re-entrant since V2.2 !

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 !

Nevertheless for example it is possible to decode a picture at the same time an other is been encoded, due to the fact, that the decoding routines do not share any data or variables with the encoding routines.

History :

~~~~~

V2.12 (22.07.1994) :

- fixed bad bug, which may have occurred when writing GIFs : one sub-routine of the SVO\_Write() function is not fully re-entrant, that's nothing new (it is protected against being called twice at a time, anyway).  
The actual bug was, that for the second, third ... n-th call not all of the global static variables had been re-initialized again. This may have caused badly written GIFs with a high likelihood. Nevertheless there was a small chance to write more than one correct GIF.  
Anyway : This is now fixed !!
- fixed small bug, which yet only occurred once :  
GIFs may contain various "extension" blocks, which do not really have a defined length. This "dynamic length" feature was only correctly implemented for "comment" blocks, so that e.g. a big-sized "application" block would have caused the program to be kept in a never-ending loop (with a high likelihood).  
This small bug is now fixed.



Maybe you've never been confronted with it.

V2.11 (13.07.1994) :

- little improvements on LZW-decoding

V2.10 (05.06.1994) :

- SVObject flags now reflect, that GfxBuffers can be written

V2.9 (03.06.1994) :

- again did some optimization of LZW decoding routines :
  - replaced some weird loops with simple bit-shifting operations
  - forced longword-usage for 68030 version(there's still room for more optimization, but can you tell me, HOW this can be done without producing garbage ... ?!)
- version still reflected 2.7 instead 2.8. Now we're at 2.9 :-)

V2.8 (27.05.1994) :

- now saving of GfxBuffers is possible : internal write routines have been completely rewritten, so that supplied Screens are transferred to GfxBuffers before saving. BitPlane AND OnePlane Buffers upto 8 Bit are supported : they are converted to each other by using the appropriate functions of superviewsupport.library V3+.

V2.7 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1  
This saves some space again (here : ca. 1600 Bytes).

V2.6 (16.04.1994) :

- now using "libcall" pragmas in 68030-Code, which should be faster
- strongly optimized LWZ decoding routines : these will hopefully be much faster on 68000 machines. On the 68030 it's hard to decide, whether there's an increase or not.
- interlaced and simple raster decoding has been joined to one single routine, so that there are no longer any unnecessary redundancies
- side effects of these changes :
  - interlaced graphics should be read somewhat faster, because the "skipping code" for the passes has been changed to allow simple adaption of the raster file decoding routine
- shortly : it's smaller (some routines), faster, better !

V2.5 (27.03.1994) :

- removed internal SVDriver-Replacement, so that we now are really
-

dependent on external SVDivers in libs:svdrivers.

This saves about 1600 Bytes disk space and reduces redundancy.

- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one. This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit more than 68030-codesize, but increases speed on 68020+ systems (should also decrease codesize, but does not ...)

V2.4 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine

V2.3 (17.03.1994) :

- fixed "large memory usage" bug in internal ECS-ChunkyPixel routine
- added several security checks (zero-pointers)
- multiple encoding was not enabled (although possible, since the decoding routines are fully re-entrant since V2.2) due to an internal flag check : now they are. This will perhaps not be fast at all, but for 4-Color GIFs it should suffice ... ;)
- improved (hopefully) LWZ-Decoding a little bit
- improved handling of "worst case" : unexpected end of file situations (source buffer for GIF Encoding too small) Now will be handled faster.
- changed 68020 version to 68030 version : both processors are binary compatible, but perhaps there are differences in execution time, which may be important for optimizations ...

V2.2 (13.03.1994) :

- decoding routines now are fully re-entrant (no more local static variables inside decoding routines)
- AGAIN : fixed same bug as in V1.2 (GIF87a ColorDepth) (re-occured in V2.1)
- this version has been released as bug-fix to V2.1 (superview.library V3.1 release)

V2.1 (09.03.1994) :

- now with SVDriver-Support. Not yet necessarily needed.
- added Write Support for GIF 87a and GIF 89a under usage of `_strongly_` modified FBM routines (see copyright notices under credits)
- improved internal "Chunky Pixel to BitMap" (8 Bit) Routine
- optimized GIF decoding a little bit !?
- fixed small bug in memory management routines
- internal routines : AUTOSCROLL now really works with big-sized Screens
- fixed several little bugs

V1.4 (29.01.1994) :

---

- fixed bug with reading of interlaced GIFs
- fixed bug in former optimizations
- small speed improvements
- added special 68020+ version (GIF.svobject.020) to the distribution archive

V1.3 (24.01.1994) :

- fixed bug with setting ColorMap under V37/V38

V1.2 (16.01.1994) :

- fixed bug with GIF87a inconsistence with global and local colordepth

V1.1 (02.01.1994) :

- first version

## 1.20 BMP.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 2.8  
Release Date : 11.06.1994

Description

~~~~~

BMP.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with ChunkyPixel-Support.

It supports reading and writing of Windows V3.00 BMP Files.
In detail these are :

Reading :

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

Writing :

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

Source Colors	Version	Type	Destination Colors
---------------	---------	------	--------------------

2	BMP V3.00	packed chk. pix.	2
4 .. 16	BMP V3.00	packed chk. pix.	16
32 .. 256	BMP V3.00	chunky pixel	256
(24 Bit)	BMP V3.00	RGB pixel	(24 Bit)

History

~~~~~

V2.8 (11.06.1994) :

- added code for reading and writing of (unpacked) 24 Bit Files
- added code for reading RLE-encoded 8 Bit files (RLE8).  
(29 of 30 pictures caused no problems, and the last one might not have been written correctly ?!)
- all BMP files are now written line-padded (longword-boundaries), if necessary and all line-padded files should now be read correctly (even more correctly than before :-)
- (hey : V2.7 was written on 05.06.94, not 06.05.94 ... )

V2.7 (05.06.1994) :

- SVObject flags now reflect, that GfxBuffers can be written

V2.6 (27.05.1994) :

- now saving of GfxBuffers is possible : internal write routines have been completely rewritten, so that supplied Screens are transferred to GfxBuffers before saving. BitPlane AND OnePlane Buffers upto 8 Bit are supported : they are converted to each other by using the appropriate functions of superviewsupport.library V3+.

V2.5 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1  
This saves some space again (here : ca. 1600 Bytes).
- removed some unused variables in write-routine

V2.4 (27.03.1994) :

- removed internal SVDriver-Replacement, so that we now are really dependent on external SVDivers in libs:svdrivers.  
This saves about 1600 Bytes disk space and reduces redundancy.
- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one.  
This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit, but increases speed on 68020+ systems  
(should also decrease codesize, but does not ...)

V2.3 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine
- there occurred some bugs with graphics, which had a width, that did not automatically implied longword-alignment :  
Now the inserted pad-bytes are read correctly for those files.

V2.2 (17.03.1994) :

- fixed "large memory usage" bug in internal ECS-ChunkyPixel routine
- added several security checks (zero-pointers)
- fixed bug in reading and writing of "Black & White" (2 Colors) BMP-Files, also changed docs (and revision notes) as far as affected by the bug-fix
- note, that 4 Color BMP Files - following to the specifications they do not exist - will neither be read nor be written (8; 2:2:2:2)
- cleaned up some garbage in this doc-file ;)

V2.1 (09.03.1994) :

- now with SVDriver-Support. Not yet necessarily needed.
- improved internal "Chunky Pixel to BitMap" (8 Bit) Routine
- fixed some small bugs in reading : handling ColorMaps handled correctly now
- writing routine(s) :
  - added code for writing 2 .. 256 Color graphics
  - 2 Colors are written as packed chunky pixel (8; 1:1:1:1:1:1:1:1)
  - 16 Colors are written as packed chunky pixel (8; 4:4)
  - 256 Colors are written as chunky pixel (8; 8)
  - 4, 8, 32, 64 or 128 Colors are expanded to either 16 or 256 Colors before writing
- internal routines : AUTOSCROLL now really works with big-sized Screens
- fixed small bug in memory management routines
- fixed several little bugs

V1.2 (26.01.1994) :

- fixed code for recognition of "not supported" (future or 24-Bit) Files

V1.1 (24.01.1994) :

- first version

## 1.21 FBM.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany  
(Also see notes under "Credits".)

FREWARE. All rights reserved.

---

Version : 2.7  
Release Date : 05.06.1994

#### Description

~~~~~

FBM.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with ChunkyPixel-Support.

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

Reading :

FBM (Unix) upto 256 Colors (8 Bit), either with Colortable or grey-scaled (selfgenerated).
Black & White with physbits=1 not supported.

Writing :

FBM (Unix) upto 256 Colors (8 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

~~~~~

V2.7 (05.06.1994) :

- SVObject flags now reflect, that GfxBuffers can be written

V2.6 (27.05.1994) :

- now saving of GfxBuffers is possible : internal write routines have been completely rewritten, so that supplied Screens are transferred to GfxBuffers before saving. BitPlane AND OnePlane Buffers upto 8 Bit are supported : they are converted to each other by using the appropriate functions of superviewsupport.library V3+.
- besides : THIS time, the history reflected V2.4 instead of V2.5 ...

V2.5 (07.05.1994) :

- use of SAS/C V6.51
  - use of new "superviewsupport.library" V1  
This saves some space again (here : ca. 1600 Bytes).
  - version still reflected 2.3 instead 2.4. Now we're at 2.5 :-)
-

V2.4 (16.04.1994) :

- improved internal "grey-scale-generation"-routine.  
It produced sometimes wrong values for the last GreyScale-Color, which should have been real "white".  
The wrong values were like this :
  - 4 GreyScales, 4. Color : 192/192/192
  - 16 GreyScales, 16. Color : 240/240/240
  - 255 GreyScales, 16. Color : 255/255/255This is now fixed. Side-effect : the grey-value below "white" now has a bigger distance to it than to the previous value (2\*256/colornum instead of 256/colornum).

V2.3 (27.03.1994) :

- removed internal SVDriver-Replacement, so that we now are really dependent on external SVDrivers in libs:svdrivers.  
This saves about 1600 Bytes disk space and reduces redundancy.
- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one.  
This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit, but increases speed on 68020+ systems  
(should also decrease codesize, but does not ...)

V2.2 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine
- now also writes FBM (Unix) upto 256 Colors (8 Bit).
- fixed ColorMap-generation for grey-scaled pictures with less than 8 Bit Depth (256 grey-scales) : values had been too small (too dark)

V2.1 (19.03.1994) :

- first version, not yet with write-support

## 1.22 C64.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany  
(Also see notes under "Credits".)

FREEWARE. All rights reserved.

Version : 2.5  
Release Date : 07.05.1994

Description

~~~~~

C64.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with ChunkyPixel-Support.

It supports reading of C64 Graphics Files (Koala, Doodle).
In detail these are :

Reading :

Format	Dimensions	Colors	Displayed as	FileSize	Flexibility
Doodle	320x200	2/16	320x200	9218	(9200 < fs < 9230)
Koala	160x200	4/16	320x200	10003	(fs == 10003)

Writing :

(not supported yet)

History

~~~~~

V2.5 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1  
This saves some space again (here : ca. 1100 Bytes).

V2.4 (27.03.1994) :

- removed internal SVDriver-Replacement, so that we now are really dependent on external SVDrivers in libs:svdrivers.  
This saves about 1600 Bytes disk space and reduces redundancy.
- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one.  
This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit, but increases speed on 68020+ systems  
(should also decrease codesize, but does not ...)
- reconstructed revision notes for V2.2 and V2.3, which were mixed together in V2.3

V2.3 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine
- removed some dead code and variables in decoding routine

V2.2 (20.03.1994) :

- slightly changed recognition routine
- and more



V2.1 (17.03.1994) :

- first version with Koala and Doodle support

## 1.23 IMG.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEMWARE. All rights reserved.

Version : 2.7  
Release Date : 15.05.1994

### Description

~~~~~

IMG.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and 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 Colors)
- IMG (18 Byte Header) GreyScaled (1..8 Bits = 2..256 Colors)

Writing :

(not supported yet)

History

~~~~~

V2.7 (15.05.1994) :

- fixed version history (V2.5 mentioned twice)
- fixed subtype strings (one blank too much)

V2.6 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1  
This saves some space again (here : ca. 1100 Bytes).
- removed some dead code

V2.5 (16.04.1994) :

---

- added support for "more than 2 GreyScales"-IMG-Files.  
Tests with 4- and 16-GreyScales have been successful, so any other color-depth should also work.
- removed some "dead" code
- removed (deactivated) debugging code
- this version uses the new "grey-scale-generation"-routine, like some of the other SVOjects written by me.  
Upto V2.4 this was not necessary, because there were only two "Colors" possible : Black and White.

V2.4 (02.04.1994) :

- improved decoding routines slightly

V2.3 (27.03.1994) :

- removed internal SVDriver-Replacement, so that we now are really dependent on external SVDrivers in libs:svdrivers.  
This saves about 1600 Bytes disk space and reduces redundancy.
- replaced SAS/C's sprintf()-routine with own, RawDoFmt()-based one.  
This saves again about 1800 Bytes.
- now using Utility-Library for integer operations (activated SAS/C's UTILLIB option) : this increases 68000-codesize a little bit, but increases speed on 68020+ systems  
(should also decrease codesize, but does not ...)

V2.2 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine
- V2.1 was released additionally to the superview.library's V3.2 developer release, V2.2 is now released inside the main package
- "#ifdef"-disabled some debugging code, which was never executed

V2.1 (23.03.1994) :

- first version

## 1.24 TIFF.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany  
(Also see notes under "Credits".)

FREWARE. All rights reserved.

Version : 2.7  
Release Date : 05.06.1994

---

## Description

~~~~~

TIFF.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with Bitplane-/ChunkyPixel-Support.

It supports reading and writing of TIFF-Files (V5.0).

In detail these are :

Reading :

Compression	Code	PixelBits	Planes	Max. Depth	As	Test
None	1	1	(system)	1*(system)	BITPLANE	
None	1	8	1,3	8 / 24	ONEPLANE	X
Mac PackBits	32773	1	(system)	1*(system)	BITPLANE	X
Mac PackBits	32773	4	1	4	ONEPLANE	X
Mac PackBits	32773	8	1	8	ONEPLANE	
LZW	5	8	1,3	8 / 24	ONEPLANE	(X)
CCITT 1D	2	1	1	1	BITPLANE	

- CCITT 1D DISABLED IN THIS VERSION : does not seem to work correctly -

Not all of these modes may work correctly and/or have been tested for working correctly. Tested were :

- None , PixelBits 8x1, ONEPLANE (problems with odd-width pics)
- None , PixelBits 8x3, ONEPLANE (problems with odd-width pics)
- Mac PackBits, PixelBits 1x?, BITPLANE (nearly no problems recognized)
- Mac PackBits, PixelBits 4x1, ONEPLANE (no problems recognized)
- LZW , PixelBits 8x3, ONEPLANE (not SURE, if working corr.)

Writing :

Writing is only supported for Graphics upto 256 Colors.

They are always written as : Motorola (MM), None, 8, 1, ONEPLANE

Correctness of the written files has been tested so far.

(Used an other program to convert the written file into IFF-ILBM and then compared the results : Just OK !)

Remarks

~~~~~

This SVObject was quite a hard piece of work and is still perhaps not fully free of bugs yet.

## A Word About ...

-----

... pictures with more than one Strip of Data :

They should be read correctly now, but I can only hope that the pictures have the same opinion ... ;-(

... LZW compressed pictures :

You may feel lucky, if decoding really works.  
If it does not, let me tell you, that I only had two pictures for testing the routines : The first one produced a strange looking picture, but decompression returned no error (8 strips in the picture). The second made the decompression routines complaining about codesizes (maximum codesize is 12 Bit, but there was a bigger one).  
I didn't found any other programs (even for MS-DOS), which were able to display the content of my test-pictures ...

#### Supported

-----

The following should work almost always :

- reading of Intel- (II) and Motorola- (MM) Files
- reading of 256 Color (8 Bit) Files,  
either packed with : - None  
                          - Mac PackBits
- reading of uncompressed 24 Bit files (RGB 3x8 Bits)

#### Not supported

-----

Files with unknown compression types (e.g. CCITT Fax) are as well rejected as files with unsupported Compression/PixelBits/Plane combinations, like e.g. None/4/2.

#### Requests and Suggestions

-----

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  
- at least `_try_` - to improve this program to also allow reading of these.

#### Debugging

-----

Nevertheless usually information on a specific picture can be requested, although the picture itself cannot be displayed.  
Not all applications may support this (e.g. SuperView does not), because the error return value may prevent them against doing this.

#### History

~~~~~

V2.7 (05.06.1994) :

- fixed a bug-fix of a bug-fix bug-fix (or such a thing :-)
ColorMaps of "II"-type files would always have been black, while ColorMaps of "MM"-type files would have been correct.
Now they are both OK (forgot conversion of II-Words).
- SVOobject flags now reflect, that GfxBuffer can be written

V2.6 (03.06.1994) :

- MAJOR REVISION

- now reads the same files as before and many, many more ... :-)
- reading of "MM" files did not work correctly, because I only checked this with self-written files (and those had been written wrong :-(. So the complete UBYTE-, UWORD- and ULONG reading stuff for TIFF's variable length parameters has been written new.
- writing of files did not work correct because "MM" UWORDS and UBYTES were not shifted into the upper word before writing.
Also fixed some minor problems.
- files without ColorMap have been reported to be "24 Bit Files". Fixed.
- reduced (temporary) memory usage for compressed pictures, where the size of the compressed data often is nothing against the maximum possible size (depending on width, height and depth).
Now we compare our calced value against the size of the whole file, which actually should be the limit ...
- added decompression routines for : LZW (Code 5) and CCITT3 (Code 2).
Although these have been derived from the original Aldus Code as distributed via the Nets, they may not work correctly at all, since I had to do many changes to get it working on the Amiga ...
(Ever ported some code from MS-DOS ? Oh, oh ...)
Remark : CCITT disabled in this version !
- added support for pictures with "more than one strip of data"
(may not work with all types of compressed data : CCITT3 decompression routines will e.g. reject data which consists of more than one strip)
- now we assume, that files with a "PhotometricInterpretation" of 2 (RGB) have at least 3 "SamplesPerPixel", if nothing else is defined :
I have seen 24 Bit RGB files, which did not contain such information !!

V2.5 (27.05.1994) :

- now saving of GfxBuffers is possible : internal write routines have been completely rewritten, so that supplied Screens are transferred to GfxBuffers before saving. BitPlane AND OnePlane Buffers upto 8 Bit are supported : they are converted to each other by using the appropriate functions of superviewsupport.library V3+.

V2.4 (07.05.1994) :

- 15.05.94 : modified Doc-File slightly
- use of SAS/C V6.51
- use of new "superviewsupport.library" V1
This saves some space again (here : ca. 1600 Bytes).

V2.3 (01.05.1994) :

- with 24 Bit pictures, PixelBits-Value in GfxBuffer was not set to 24 instead of 8 accidentally. Fixed.
- version still reflected 2.1 instead 2.2. Now we're at 2.3 :-)
- "black & white" graphics without "BitsPerSample" and "SamplesPerPixel" Tags should now be read correctly instead of rejecting them
(default value is "1" for both)
- still working on LZW-decoding (and working, and working, and working, and working, ... :-)

V2.2 (18.04.1994) :

- improved speed of MacPackBits decoding routines
- fixed bug, which occurred, when any BitPlane-Formats (no ChunkyPixel) had been read : Black Screen appeared. This is now fixed.
- created new hierarchy for decoding routines : they are now much more flexible and error-resistant. Future expansions will be easier to make.
- temporary buffer for rawdata is now delocated immediately after use
- reduced memory usage for BitPlane Formats a lot
- added code for reading 24 Bit Data (unpacked 8 Bit RGB)
- reduced memory usage for unpacked 8 Bit-ChunkyPixel/RGB by just taking the rawbuffer, which then hasn't to be given free anymore : no copying is done any longer !
- improved internal "grey-scale-generation"-routine.
It produced sometimes wrong values for the last GreyScale-Color, which should have been real "white".
The wrong values were like this :
 - 4 GreyScales, 4. Color : 192/192/192
 - 16 GreyScales, 16. Color : 240/240/240
 - 255 GreyScales, 16. Color : 255/255/255
 This is now fixed. Side-effect : the grey-value below "white" now has a bigger distance to it than to the previous value (2*256/colornum instead of 256/colornum).

V2.1 (02.04.1994) :

- first version, not yet fully free of bugs - I guess :-(- and not supporting the full set of features possible

1.25 Targa.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 2.4
Release Date : 26.06.1994

Description
~~~~~

Targa.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with Bitplane-/ChunkyPixel-Support.

It supports reading of TGA (Truevision Targa) Files.  
In detail these are :

Reading :

- monochrome Targa-Files (1 Bit, planar)
- Targa with 256 Colors (8 Bit)  
(files without ColorMap are displayed in grey-scales)
- Targa 24 Bit-Files

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 BGR | 256<br>(24 Bit)    |

Remarks

~~~~~

- Currently files with Colorbits other than 1, 8 or 24 (16, 32) are not supported yet.
- ColorMaps have to be of type "3-Byte RGB".
- RLE-encoded files are rejected yet.
- The flags for "mirroring" Images vertically and/or horizontally are not fully interpreted yet, nevertheless they are checked and reported via 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

~~~~~

V2.4 (26.06.1994) :

- fixed small bug, which might have caused this one crashing/rejecting, when writing from a Screen (instead of a GfxBuffer).

V2.3 (12.06.1994) :

- now also writes Targa files (all input converted to either 8 or 24 Bit)
- now also reads monochrome pictures with pixelbits value of 1 (planar)
- files with no VERTINV set are now converted (top becomes bottom)
- changed way of file-recognition : files with unknown flags are now read as if these flags would not have been set. This may cause problems with files, which depend on correct interpretation of those flags, but often they are written wrong, so that this might have caused files to be rejected. Now we check an other Byte-combination for plausibility reasons.

V2.2 (22.05.1994) :

---

- removed some unused debugging code : this was accidentally called always, when a picture has been loaded. This did not crash the program, but always caused a requester, which had to be confirmed by the User.

V2.1 (22.05.1994) :

- first version

## 1.26 WPG.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 2.4  
Release Date : 22.05.1994

### Description

~~~~~

WPG.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with Bitplane-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

Writing :

(not supported yet)

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, this will be displayed, 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 FileInfoRequester.

---



This version actually has been tested with graphics with 1, 4 and 8 Bit ColorDepth (2, 16 and 256 Colors).  
Due to the fact, that the 2 Bit-Routine is identically to the 1 Bit-Routine should not get any problems with those pictures.

History  
~~~~~

V2.4 (22.05.1994) :

- fixed bug, which caused the last 4 Bytes of the ColorMap not to be read (last two Colors have been wrong, then).
This did not concern grey-scaled Files without own ColorMaps ;-)
- fixed the same bug for BitMap data : the last 10 Bytes might not have been read.
- modified Doc-File slightly to reflect, that 8 Bit-Files are read correctly (not just since V2.4 ...)

V2.3 (15.05.1994) :

- fixed subtype strings (one blank too much)
(better than nothing, eh ?)

V2.2 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1
This saves some space again (here : ca. 1100 Bytes).
- version accidentally reflected 2.5 instead 2.1. Now we're at 2.2 :-)
- removed some dead code

V2.1 (17.04.1994) :

- first version

1.27 SunRaster.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 2.1
Release Date : 26.06.1994

Description
~~~~~

SunRaster.svobject is an external Library-Module for the

---

superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with Bitplane-/ChunkyPixel-Support.

It supports reading 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

~~~~~

- Currently files with Colorbits other than 1, 8 or 24 are not supported yet.
- only RGB-plane ColorMaps supported (or monochrome, without map)
- max. 1 plane of BitMap data allowed
- Loading of 24 Bit files has not been tested yet, but should work.

History

~~~~~

V2.1 (26.06.1994) :

- first version

## 1.28 Pictor.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 2.1  
Release Date : 13.06.1994

Description

~~~~~

Pictor.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with Bitplane-/ChunkyPixel-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).

Writing :

(not yet supported)

History

~~~~~

V2.1 (13.06.1994) :

- first version

## 1.29 MAC.svobject

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 2.1  
Release Date : 12.05.1994

Description

~~~~~

MAC.svobject is an external Library-Module for the superview.library V2+.

It contains SVDriver-Support for the superview.library V3+, and needs any SVDriver with Bitplane-Support.

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

Reading :

- MAC Black & White 576x720

Writing :

(not supported yet)

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. Due to the fact, that we can't just check the suffix or prefix of the file to be "MAC", we have to check the filetype also.

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 !

## History

~~~~~

V2.1 (12.05.1994) :

- first version

1.30 JPEG.svobject

The "JPEG.svobject" can be found in the SVOJPEG23.lha" Archive File.

The documentation and all other distribution files can be found in this archive.

1.31 ECS.svdriver

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 1.9

Release Date : 05.09.1994

Requirements

~~~~~

- OS V2.04+ (V37)+ and its libraries
- superviewsupport.library V1+
- AMIGA with Enhanced Chip Set (ECS)

## Description

~~~~~

ECS.svdriver is an external Library-Module for the superview.library V3+.

It supplies functions for V2+ SVOjects to allow object-orientated displaying of graphics on ECS Screens.

The internal graphics interchange format is the same as supported by superview.library V3+ in connection with V2+ SVOjects, which can be used by external custom programs.

ECS.svdriver is identical to AGA.svdriver, except to the "chunky pixel" support and other features, which have been realized without V39 functions here.

This Driver supports the following :

Dimensions	Depth	Type	
[ECS]	[ECS]	BITPLANE	
[ECS]	8	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.

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

Known Bugs

~~~~~

There's a bug in the chunky pixel routines of OS V37-39, which has been fixed (work-around), in a way, that the last (upto) 7 pixels of a ONEPLANE/chunky graphics, which has a width ending not on a byte boundary (as e.g. 433) will be cut off.

That's not really a good solution, but better than crashing the whole system.

Maybe this bug will be fixed in future OS or chipset versions.

#### History

~~~~~

V1.9 (05.09.1994) :

- added work-around for chunky-pixel bug in the OS (see library-docs for V8.2)

V1.8 (11.06.1994) :

- fixed small bug in ColorMap routine : creation of RGB-values resulted in a mask of 0xFFFFFFFF00 instead of 0xFFFFFFFF (although there's not really a mask used) (V39 only)

V1.7 (07.05.1994) :

- use of SAS/C V6.51

- use of new "superviewsupport.library" V1
This saves some space again (here : ca. 400 Bytes).

V1.6 (30.04.1994) :

- fixed small bug in memory delocation (when multiply called)

V1.5 (02.04.1994) :

- improved internal "regular case" BitMap-to-BitMap routine

V1.4 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine
- We now use a higher alignment for TmpRas-BitPlanes,
to ensure higher compatibility (and speed increase) under
V39 with the AGA chipset.
Alignment now is done for longword-boundaries as follows :
 $((width + 31) \gg 5) \ll 5$

V1.2 / V1.3 (14.03.1994) :

- there are no significant differences between V1.2 and V1.3
(at least I don't remember them ...)
- fixed "large memory usage" bug in ChunkyPixel routine
- this version has been released as bug-fix to V1.1
(superview.library V3.1 release)

V1.1 (09.03.1994) :

- first version

1.32 AGA.svdriver

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 1.10
Release Date : 05.09.1994

Requirements

~~~~~

- OS V3.00+ (V39)+ and its libraries
- 68020+
- superviewsupport.library V1+
- AMIGA with AGA ChipSet (ECS still supported, but with restrictions)

## Description

~~~~~

AGA.svdriver is an external Library-Module for the superview.library V3+.

It supplies functions for V2+ SVOjects to allow object-orientated displaying of graphics on AGA Screens.

The internal graphics interchange format is the same as supported by superview.library V3+ in connection with V2+ SVOjects, which can be used by external custom programs.

AGA.svdriver is identical to ECS.svdriver, except to the "chunky pixel" support and other features, which have been realized with V39 functions here.

This Driver supports the following :

Dimensions	Depth	Type	
[AGA]	[AGA]	BITPLANE	
[AGA]	8	ONEPLANE	(Chunky Pixel)

ONEPLANE 24 Bit Graphics (RGB 3x8-ONEPLANE) will be dithered to 8 Bit.

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

Known Bugs

~~~~~

There's a bug in the chunky pixel routines of OS V37-39, which has been fixed (work-around), in a way, that the last (upto) 7 pixels of a ONEPLANE/chunky graphics, which has a width ending not on a byte boundary (as e.g. 433) will be cut off.

That's not really a good solution, but better than crashing the whole system.

Maybe this bug will be fixed in future OS or chipset versions.

## History

~~~~~

V1.10 (05.09.1994) :

- added work-around for chunky-pixel bug in the OS
(see library-docs for V8.2)

V1.9 (29.06.1994) :

- now attaches ViewPortExtra structure to Screens, if possible, so that hopefully display-speed will increase
- now opening interleaved Screens, when displaying ONEPLANE graphics for speed reasons
- also, no longer intermediate CopperList-Updates are allowed for the display screen
- one of the optimizations done in V1.8 might have caused slowe 24 Bit display instead of faster (not sure). Now we go the safe way,

but still somewhat optimized.

V1.8 (11.06.1994) :

- now at least requires 68020 CPU
(there are perhaps no AGA machines out there, which have none).
This should speed up the 24 Bit "dithering" routines.
Startup-Code is still compatible to 68000, so that an installation on 68000 systems will not cause a crash, if running V37/38.
- made all variables 32 Bit-wide
- did some optimizations
- fixed small bug in ColorMap routine :
creation of RGB-values resulted in a mask of 0xFFFFFFFF00 instead of 0xFFFFFFFF (although there's not really a mask used)
- removed some V37-code, which was never executed

V1.7 (22.05.1994) :

- fixed small bug in 24-Bit-Support :
our single temporary bitplane line always had a PlaneDepth of 24,
so that 16 planes were wasted useless :-(
Now this memory is saved and also speed may increase.

V1.6 (07.05.1994) :

- use of SAS/C V6.51
- use of new "superviewsupport.library" V1
This saves some space again (here : ca. 400 Bytes).
- version still reflected 1.4 instead 1.5. Now we're at 1.6 :-)

V1.5 (30.04.1994) :

- fixed small bug in memory delocation (when multiply called)

V1.4 (18.04.1994) :

- improved internal "regular case" BitMap-to-BitMap routine
- added code for dithering 24 Bit RGB-Oneplane pictures
(is quite slow, but results in good pictures)

V1.3 (25.03.1994) :

- slightly improved internal BitMap-to-BitMap routine
- removed bug-fix from V1.2, now using BMF_DISPLAYABLE flag, no matter if this will help or not

V1.2 (19.03.1994) :

- tried to fix bug with strange-sized (temporary) bitmaps like 694x136 :
now uses size = ((x + 15) >> 4) << 4) for determinating the
-

desired width.

This is not completely fixed yet (seems to be an OS bug ...)

V1.1 (09.03.1994) :

- first version

1.33 EGS.svdriver

© 1994 by Andreas R. Kleinert, Grube Hohe Grethe 23, 57074 Siegen, Germany

FREEWARE. All rights reserved.

Version : 1.7
Release Date : 16.07.1994

Requirements

~~~~~

- OS V2.04+ (V37)+ and its libraries
- superviewsupport.library V3+
- egs.library V1+
- egsintui.library V1+
- egsgfx.library V1+
- AMIGA with EGS Graphic Card (or EGS distribution with Amiga Emulation)

#### Description

~~~~~

EGS.svdriver is an external Library-Module for the superview.library V3+.

It supplies functions for V2+ SVOjects to allow object-orientated displaying of graphics on EGS Screens (Enhanced-Graphic-System Screens). All corresponding EGS-Libraries are needed.

The internal graphics interchange format is the same as supported by superview.library V3+ in connection with V2+ SVOjects, which can be used by external custom programs.

This Driver supports the following :

Emulation	Dimensions	max. Depth	Type
EGS-Card	[EGS-Card]	8/24	ONEPLANE
		8	BITPLANE
ECS/AGA	[ECS/AGA]	8	ONEPLANE
		8	BITPLANE

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.

EGS-Cards

When using an EGS Card, all supported ColorDepths should be displayed correctly in TrueColor.

Amiga ECS/AGA emulation

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 ...

Closing of the current EGS Display is possible by freeing the handle's resources or delocating the handle, but no screen or window addresses (for IDCMP checking) are returned by the related library functions, because EGS is (as far as I know) binary incompatible to Intuition. To fix this a little bit, every time when displaying a graphic on the EGS-Screen a Workbench Window is opened which allows usual IDCMP access. Window IDCMP and Window Flags (except backdrop/borderless) are recognized and used.

History

~~~~~

V1.7 (16.07.1994) :

- when EGS.svdriver could not be opened, superviewsupport.library (and some others) would not have been closed (and thus could not be flushed out of memory anymore).  
This is now fixed (and still has to be fixed for some other SVOjects and SVDdrivers).

V1.6 (22.05.1994) :

- replaced internal BitPlaneToOnePlane routine with its pendant from superviewsupport.library V3+

V1.5 (07.05.1994) :

- 15.05.94 : modified Doc-File slightly
  - use of SAS/C V6.51
  - use of new "superviewsupport.library" V1  
This saves some space again (here : ca. 400 Bytes).
-

V1.4 (29.04.1994) :

- MAJOR REVISION
- added support for BITPLANE formats upto 8 Bit (256 Colors), which works for V37+ and V39+ (different routines)
- fixed old bug in memory delocation routines (for temporary buffers), which had not been used between V1.2 - V1.3, thus this bug might obviously have been the reason, why V1.1 crashed with some mysterious Exec-Gurus sometimes.

V1.3 (18.04.1994) :

- MAJOR REVISION
- changed way of setting/handling Colors : now using ObtainColor() and ReleaseColor(). Colors are released immediately after writing the graphics into the BitMap, to allow any later displayed graphics to obtain their own colors : otherwise no more colors would be free and they would be displayed with the wrong colors.
- improved handling of 8 Bit graphics :
  - ECS/AGA emulation :  
256 Color graphics usually will be displayed with some wrong colors, which are actually obtained for the EGS-Display itself (Window Borders, etc.) and cannot be changed by the application.  
ChunkyPixel-Graphics with less than 256 Colors will be displayed 100% perfect as long as only unused Color-Registers are already obtained by any applications.
  - EGS Cards :  
Due to the fact, that we are working with 24 Bit, all graphics should be displayed with their 100% correct Colors.
- added support for 24 Bit graphics :
  - ECS/AGA emulation :  
24 Bit graphics are dithered to 256 Colors and should be displayed in GreyScales usually (see notes concerning 256 Color graphics)
  - EGS Cards :  
24 Bit graphics are displayed in TrueColor

V1.2 (29.03.1994) :

- small improvements :
  - now using SMART\_REFRESH Window
  - no longer Backdrop-Window
  - fixed handling of Control-Window
- with newer versions of the EGS-Libraries it should no longer crash (tested with Version 15-Mar-1993, found on AmigaMagazin PD 4/93-4a/b)

V1.1 (17.03.1994) :

---

- first version with the beginnings of rudimentary EGS-Support

## 1.34 Used literature for developing 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] ... and perhaps books/magazines/articles, which I don't remember yet !

1.35 Credits

Credits:

~~~~~

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).

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" :

```
* 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)
*
```

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 V6.51).

(Actually I'm not sure, if it really works ...)

In detail, I'm making use of strongly modified versions of the lzw  
(lzwdec.c) and CCITT 1D (tiff2.c) decoding routines.

### 1.36 Requirements for the SuperView.library Package

- OS V2.04+ (V37)+ and its libraries,  
as e.g. iffparse.library V37+ (only for the ILBM/ACBM.svobject)
- for DataType support under OS V39+ you need "datatypes.library" V39+  
(not necessary, if no DataType-Support is needed)
- xpkmaster.library V2+ (only for the SVO.svobject)  
which is (C)opyright by its authors (c/o Urban Dominik Mueller)
- superviewsupport.library V1+, V2+, V3+ or V4+,  
which depends on the specific SVOjects / SVDdrivers  
This distribution always includes the latest version of the library.
- some special requirements for the SVDdrivers (read it there!)