

super_picture

COLLABORATORS

	<i>TITLE :</i> super_picture		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 14, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	super_picture	1
1.1	Amiga-E MODULE: Super_Picture	1
1.2	Amiga Foundation Classes: Super_Picture / History	3
1.3	Author(s) Info	4
1.4	Introduction	4
1.5	Super_Picture / Error Table	5
1.6	Amiga Foundation Classes: Super_Picture/Super_Picture()	5
1.7	Amiga Foundation Classes: Super_Picture/clear()	5
1.8	Amiga Foundation Classes: Super_Picture/load()	6
1.9	Amiga Foundation Classes: Super_Picture/check()	7
1.10	Amiga Foundation Classes: Super_Picture/paltoscr()	7
1.11	Amiga Foundation Classes: Super_Picture/bmhd()	7
1.12	Amiga Foundation Classes: Super_Picture/bitmap()	8
1.13	Amiga Foundation Classes: Super_Picture/numcols()	8
1.14	Amiga Foundation Classes: Super_Picture/modeid()	8
1.15	Amiga Foundation Classes: Super_Picture/width()	8
1.16	Amiga Foundation Classes: Super_Picture/height()	9
1.17	Amiga Foundation Classes: Super_Picture/depth()	9
1.18	Amiga Foundation Classes: Super_Picture/Super_Picture()	9
1.19	Amiga Foundation Classes: Super_Picture/bmobj()	10
1.20	Amiga Foundation Classes: Super_Picture/palette()	10
1.21	Amiga Foundation Classes: Super_Picture/version()	10
1.22	Amiga Foundation Classes: Super_Picture/cludge()	10
1.23	Amiga Foundation Classes: Super_Picture/Super_Picture()	11
1.24	Amiga Foundation Classes: Super_Picture/loaddraw()	11
1.25	Amiga Foundation Classes: Super_Picture/image()	12
1.26	Amiga Foundation Classes: Super_Picture/dtload()	12
1.27	Amiga Foundation Classes: Super_Picture/remap()	13
1.28	Amiga Foundation Classes: Super_Picture/trueconv()	14
1.29	Amiga Foundation Classes: Super_Picture/empty()	14
1.30	Amiga Foundation Classes: Super_Picture/trueBitmap()	14

Chapter 1

super_picture

1.1 Amiga-E MODULE: Super_Picture

** Super_Picture - Original By Fabio Rotondo **

** This Object is part of the Amiga Foundation Classes **

** DOCUMENTATION GUIDE **

Introduction

Author Info

Amiga Foundation Classes

History

Requires: IFFParser , Bitmapper , Palette
trueBitmap , resourceTracker

AFC Tools / bitmap2true

Base: \$0007000

COMMANDS

BRIEF DESCRIPTION

super_picture(pal=TRUE, resTracker)
Class Constructor.

bitmap()
Returns picture's bitmap.

bmhd()
Returns the BitmapHeader.

bmobj()
Returns embedded bitmapper.

check(filename)
Checks IF the file is a valid ILBM.

clear()

Frees ALL memory.

`cludge()`
Converts a memory area into a picture.

`depth()`
Returns picture's depth.

`dtload(fname, layers, rport, scr)`
Loads a picture using DataTypes only.

`dump()`
Dumps a picture inside a file in raw format.

`empty()`
Checks against class emptyness.

`height()`
Returns picture's height.

`image()`
Creates AND returns picture's image structure.

`load(fname, layers, rport, trap)`
Loads up a picture.

`loadraw(filename)`
Loads a dumped file into memory.

`modeid()`
Returns picture's ModeID.

`numcols()`
Returns number of colours.

`palette()`
Returns the picture's palette class.

`paltoscr(screen)`
Set the screen palette.

`rastport()`
Returns the bitmap rastport.

`remap(scr, mode=REMAP_SCREEN)`
Remaps the bitmap TO a NEW palette.

`trueBitmap()`
Return embedded trueBitmap class.

`trueconv()`
Creates the trueBitmap

`width()`
Returns picture's width.

`version()`

Returns class version AND revision.

Error Table

1.2 Amiga Foundation Classes: Super_Picture / History

HISTORY

- V2.20 - ADD: resourceTracker support
ADD:
trueBitamp()
method.
- V2.10 - ADD:
empty()
method.
- V2.00 - ADD: *great* the NEW
remap()
method!
ADD: *great* the NEW
trueconv()
method!
ADD: now the palette is handled using the Palette Class
- ENH: now the
palette()
method returns the Palette Class pointer.
- ENH: now the
load()
method will remap the picture also under OS2.04+
if you'll provide a valid screen pointer.
- V1.50 - Add:
dtload()
method
- Enh: now datatype loading supports remapping
- Enh: Added an optional parameter to the
load()
method.
- V1.40 - Add: image() method
- V1.30 - Add: dump() method
- V1.20 - Fix: small bug in decodebody() method
- V1.10 - Add: cludge() method
- V1.00 - Inital release
-

1.3 Author(s) Info

Original By: Fabio Rotondo (fsoft@intercom.it)

E Version By: Fabio Rotondo

Address:

Fabio Rotondo
C.so Vercelli 9
28100 Novara
ITALY

e-mail: fsoft@intercom.it
Fabio.Rotondo@deagostini.it

Phone: (ITA) - (0)321 - 459676 (home)
(ITA) - (0)321 - 424272 (office)
(ITA) - (0)338 - 7336477 (GSM Phone)

Fax: (ITA) - (0)321 - 424560

Web: <http://www.intercom.it/~fsoft> (my home page)
<http://www.intercom.it/~fsoft/ablast.html> (Amiga Blast Home Page) ↔

1.4 Introduction

Super Picture Class - Introduction

This is a very useful class which interfaces the programmer with OS2.0 and OS3 ↔
.0

features in loading pictures.

This class allows you to load any kind of picture (via DataTypes) without ↔
worrying
about interfacing with datatypes.library, because it does it all by itself.

The great thing is that is the class itself which knows if DataTypes are ↔
available
or not and, if not, the class will just handle standard ILBM IFF files, using ↔
a
very fast ILBM decoding routine built in: no need of iff.library or any other
"non-standard Amiga" libraries.

I hope you will enjoy this class.

Fabio Rotondo

NOTE: many thanks to Alfred P. Schwarz, who elped me in writing the image() ↔
method.

1.5 Super_Picture / Error Table

```
EXPORT ENUM SPERR_NOMEMORY=SUPERPIC_BASE, SPERR_DTLIB, SPERR_OBJCRE, ↔
          SPERR_DOMETHOD, SPERR_SAVEFILE,
          SPERR_OPENSAREFILE, SPERR_BMHD, SPERR_CMAP, SPERR_BODY
```

VALUE	CONSTANT NAME	DESCRIPTION
\$0000	SPERR_NOMEMORY	No Memory
\$0001	SPERR_DTLIB	Could not open 'datatypes.library'
\$0002	SPERR_OBJCRE	Could not create datatype OBJECT
\$0003	SPERR_DOMETHOD	DoMethod() call failed
\$0004	SPERR_SAVEFILE	Could not save file
\$0005	SPERR_OPENSAREFILE	Could not open file FOR saving
\$0006	SPERR_BMHD	BitMapHeader Chunk not found
\$0007	SPERR_CMAP	ColorMap chunk not found
\$0008	SPERR_BODY	Body Chunk not found

1.6 Amiga Foundation Classes: Super_Picture/Super_Picture()

NAME: super_picture(pal=TRUE, resTracker = NIL:PTR TO resourceTracker)

DESCRIPTION: This is the class constructor.

INPUT: pal - (DEFAULT: TRUE) this is a flag and it is used to ↔
tell the class whether you want to store also a palette ↔
along with picture data or not.
NOTE: if you set it to TRUE, additional 256*3 bytes ↔
of memory will be allocated during construction.

resTracker - (Optional) Pointer TO a valid resourceTracker class.

RESULTS: NONE.

SEE ALSO: resourceTracker

1.7 Amiga Foundation Classes: Super_Picture/clear()

NAME: clear()

DESCRIPTION: This method will free all allocated resources inside a super_picture class.

INPUT: NONE.

RESULTS: Memory will be freed.

NOTE: It is safe to call this method also if the class has not been used yet.

SEE ALSO: bitmapper / clear()

1.8 Amiga Foundation Classes: Super_Picture/load()

NAME: load(fname: PTR TO CHAR, layers=FALSE, rastport= ← FALSE, trap=TRUE, scr=NIL)

DESCRIPTION: This is the biggy. This method will try to load a picture in memory. If the user has OS3.0+, also all picture formats supported by datatypes can be loaded. Under OS2.0+ only ILBM IFF files will be supported.

INPUT: fname - The file name of the picture you wish to load. ←

layers - (DEFAULT: FALSE) If you want to create a layer related to the picture you are loading, set this to TRUE. ←

rastport - (DEFAULT: FALSE) If you want to create a rastport to the picture you are loading, set this to TRUE.

trap - (DEFAULT: TRUE) If the picture does not exist (i.e. "File not found"), then the class WILL NOT return raise any error, but you will get the error number as a return code (you can also ignore it, if you want.) If it is set to FALSE, then the class will raise an exception.

scr - (DEFAULT: NIL) You can provide a screen from where the palette will be used to remap the picture during loading.

RESULTS: if trap is TRUE and an error occurred, you will have as a result the error value.

SEE ALSO: dtload()

```

                                Bitmapper / allocbitmap()
IFFParser / load()

```

1.9 Amiga Foundation Classes: Super_Picture/check()

NAME: check(filename : PTR TO CHAR)

DESCRIPTION: This method checks the given file and says if it is a valid IFF ILBM file or not.

INPUT: filename - Name of the file to check

RESULTS: TRUE - It is a valid IFF ILBM file

FALSE - It is not a valid IFF ILBM file

NOTE: Usually, you should not use this method, because the

load()
method will do this check for you.

SEE ALSO: IFFParser / getheader()

1.10 Amiga Foundation Classes: Super_Picture/paltoscr()

NAME: paltoscr(scr:PTR TO screen, view=FALSE)

DESCRIPTION: This method sets up the palette of a screen accordingly to the picture palette.

NOTE: starting from V2.0, you can also pass a viewport as a parameter.

INPUT: scr - A *valid* screen pointer.
(or a *valid* viewport pointer)

view - (Optional) Set this to TRUE if you are providing a viewport pointer instead of a screen pointer.

RESULTS: NONE.

SEE ALSO: setPal()

1.11 Amiga Foundation Classes: Super_Picture/bmhd()

NAME: bmhd()

DESCRIPTION: This method returns the current BitMapHeader.

INPUT: NONE.

RESULTS: A pointer to a valid BitMapHeader structure.

SEE ALSO:

1.12 Amiga Foundation Classes: Super_Picture/bitmap()

NAME: bitmap()

DESCRIPTION: This method returns a pointer to the current bitmap structure.

INPUT: NONE.

RESULTS: a pointer to a Bitmap structure. It may be NIL.

SEE ALSO: bitmapper / bitmap()

1.13 Amiga Foundation Classes: Super_Picture/numcols()

NAME: numcols()

DESCRIPTION: This method returns the number of colours of the loaded picture.

INPUT: NONE.

RESULTS: number of colours of the picture. It may be 0.

SEE ALSO:

1.14 Amiga Foundation Classes: Super_Picture/modeid()

NAME: modeid()

DESCRIPTION: This method returns the default modeid of the picture.
The ModeID is the standard monitor value for displaying the picture.

INPUT: NONE.

RESULTS: a LONG value, containing the ModeID.

SEE ALSO:

1.15 Amiga Foundation Classes: Super_Picture/width()

NAME: width()

DESCRIPTION: This method returns picture's width.

INPUT: NONE.

RESULTS: a INT containing picture's width. It may be 0.

SEE ALSO:

height()

depth()

1.16 Amiga Foundation Classes: Super_Picture/height()

NAME: height()

DESCRIPTION: This method return picture's height.

INPUT: NONE.

RESULTS: a INT value containing the picture's height. It may be 0.

SEE ALSO:

width()

depth()

1.17 Amiga Foundation Classes: Super_Picture/depth()

NAME: depth()

DESCRIPTION: This method returns the picture's depth.

INPUT: NONE.

RESULTS: a BYTE value containing the picture depth. It may be 0.

SEE ALSO:

width()

height()

1.18 Amiga Foundation Classes: Super_Picture/Super_Picture()

NAME: rastport()

DESCRIPTION: This method returns a pointer to the picture's rastport.

INPUT: NONE

RESULTS: a ptr to the picture's rastport. It may be NIL.

SEE ALSO: bitmapper / rastport()

1.19 Amiga Foundation Classes: Super_Picture/bmobj()

NAME: bmobj()

DESCRIPTION: This method returns a pointer to the embedded bitmapper class.

INPUT: NONE.

RESULTS: a ptr to the embedded bitmapper class: USE IT AT YOUR OWN RISK!

SEE ALSO: bitmapper

1.20 Amiga Foundation Classes: Super_Picture/palette()

NAME: palette()

DESCRIPTION: This method returns picture's palette class.

INPUT: NONE.

RESULTS: this method has slightly changed since V1.x of Super_Picture class, since now it returns a pointer to the hidden Palette class and not the palette anymore.

SEE ALSO: Palette Class

1.21 Amiga Foundation Classes: Super_Picture/version()

NAME: version()

DESCRIPTION: This method returns class version and revision.

INPUT: NONE.

RESULTS: This method returns TWO values, version and revision.

SEE ALSO:

1.22 Amiga Foundation Classes: Super_Picture/cludge()

NAME: cludge(mem:PTR TO CHAR)

DESCRIPTION: This method converts a memory area into a bitmap.

INPUT: mem - PTR to the starting memory area.

RESULTS:

NOTE: this MUST be a `_valid_` memory area created using the
`dump()`
method.

SEE ALSO:

`dump()`
`bitmapper / cludge()`

1.23 Amiga Foundation Classes: Super_Picture/Super_Picture()

NAME: dump(filename)

DESCRIPTION: This method dumps the current picture into a file in raw format.

INPUT: filename - Name of the file where to dump the picture.

RESULTS: NONE.

NOTE: Current version of this method does not save the palette.

SEE ALSO:

`cludge()`
`bitmapper / dump()`

1.24 Amiga Foundation Classes: Super_Picture/loadraw()

NAME: loadraw(filename)

DESCRIPTION: This method loads a file created with
`dump()`
method
and creates a picture.

INPUT: filename - Name of the raw data file to load.

RESULTS: The picture will be loaded inside the object.

NOTE: All data in the object will be lost.

SEE ALSO:

`clear()`

```
dump()
bitmapper / loaddraw()
```

1.25 Amiga Foundation Classes: Super_Picture/image()

NAME: image()

DESCRIPTION: Sometimes, it is useful to obtain picture Image structure, the one ←

used by Intuition gadgets. Using this method, you can obtain this Image structure.

INPUT: NONE

RESULTS: a pointer to a valid Image structure.

NOTE: This method is *slow* the first time you call it, because it has to create the Image structure and picture. Then, it will be fastlighting, until you don't load another picture in the class.

SEE ALSO: Bitmapper / image()

1.26 Amiga Foundation Classes: Super_Picture/dtload()

NAME: dtload(fname: PTR TO CHAR, layers=FALSE, rastport ←
=FALSE, scr=NIL:PTR TO screen)

DESCRIPTION: This is a lower level method than the load() one.

Using this method you will load a picture using DataTypes ONLY. This means that you can use this only on OS3.0+.

This method is faster than the load() one because

it does not perform the usual "format checking" before trying to load the picture. ←

INPUT: fname - The file name of the picture you wish to dtload. ←

layers - (DEFAULT: FALSE) If you want to create a layer related to the picture you are dtloading, set this to TRUE. ←

rastport - (DEFAULT: FALSE) If you want to create a rastport to the picture you are dtloading, set this to TRUE.

scr - (DEFAULT: NIL) You can provide a pointer to a screen as the last parameter. If a screen pointer is passed, then the picture will be remapped accordingly to the palette attached to the provided screen.

NOTE: This method CANNOT be trapped. If an error occurred, it will raise an error.

RESULTS: NOTHING.

SEE ALSO:

load() Bitmapper / allocbitmap()
IFFParser / dtload()

1.27 Amiga Foundation Classes: Super_Picture/remap()

NAME: remap(scr:PTR TO screen, mode=REMAP_SCREEN)

DESCRIPTION: this method is used to remap the picture currently in memory to fit inside a new screen with a different palette. The remap use a quite smart routine :)

INPUT: scr - Pointer to a valid screen or viewport or palette class.

mode - (OPTIONAL) Use it to describe the remap method. You *must* provide the right value if you are passing a pointer to a viewport or a palette class. Available values are:

REMAP_SCREEN - If you passed a pointer to a screen.
REMAP_VIEWPORT - If you passed a pointer to a viewport
.
REMAP_PALETTE - If you passed a pointer to a palette class.

NOTE: In both REMAP_SCREEN and REMAP_VIEWPORT mode, the remap() method will test the destination screen/viewport depth and will arrange the bitmap to fit the destination depth. In the REMAP_PALETTE mode, the current depth is assumed.

After a remap, you should *not* perform another remap on the same bitmap, because palette RGB values are wrong.

The remap() method could use the blitter.

The `remap()` method could allocate a new bitmap.

RESULTS: the bitmap will be ready to be blitted on the new screen/viewport ↔

SEE ALSO: Palette Class
Bitmapper/assign()

1.28 Amiga Foundation Classes: Super_Picture/trueconv()

NAME: `trueconv()`

DESCRIPTION: this method will return a pointer to a new
trueBitmap class
containing the current picture in the super_picture class.

INPUT: NONE

RESULTS: a pointer to a trueBitmap class.
It may be NIL.

SEE ALSO:
trueBitmap()

1.29 Amiga Foundation Classes: Super_Picture/empty()

NAME: `empty()`

DESCRIPTION: use this method to check against class emptyness.

INPUT: NONE

RESULTS: a boolean value. FALSE means that something is inside
the class: the bitmap, the palette or the trueBitmap.

SEE ALSO: Bitmapper / empty()
Palette / empty()
trueBitmap / empty()

1.30 Amiga Foundation Classes: Super_Picture/trueBitmap()

NAME: `trueBitmap()`

DESCRIPTION: this method will return a pointer to a `_valid_`

trueBitamp
class embedded in
the super_picture class.

INPUT: NONE

RESULTS: a pointer to a valid
trueBitmap
class.
It may be NIL.

SEE ALSO:
