

**SGrab.doc**

**COLLABORATORS**

	<i>TITLE :</i> SGrab.doc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 15, 2022	

**REVISION HISTORY**

<i>NUMBER</i>	<i>DATE</i>	<i>DESCRIPTION</i>	<i>NAME</i>

# Contents

<b>1</b>	<b>SGrab.doc</b>	<b>1</b>
1.1	Table of Contents	1
1.2	What to do with it	1
1.3	Legal stuff	2
1.4	What you need to use it	2
1.5	How to install it	2
1.6	How to configure and use it	3
1.7	FILE: Specify a filename	5
1.8	NUMBER: Number files all the way through	6
1.9	WIDTH: The width of a screen of window	7
1.10	HEIGHT: The height of a screen of window	7
1.11	DEPTH: The depth of a screen	7
1.12	SIZE: The size of a screen of window	8
1.13	TITLE: The title of a screen of window	8
1.14	OWNER: The owner of a screen of window	8
1.15	TYPE:	9
1.16	TIME:	9
1.17	DATE:	9
1.18	ICON: Define an icon template	9
1.19	CX_POPKEY: Hotkey that tells SGrab to open its window	9
1.20	CX_POPUP: Open user interface when sgrab gets started	10
1.21	CX_PRIORITY: Priority of commodity handler	10
1.22	GRABSCRKEY: Hotkey that grabs the current screen	10
1.23	GRABWINKEY: Hotkey that grabs the current window	10
1.24	PUBSCREEN: Screen to open the window on	10
1.25	CLIPUNIT: Clipboard unit to save image data to	11
1.26	NUMBER: Number files all the way through	11
1.27	DELAY: Define time to wait before grabbing	11
1.28	WINDOW: Grab a window, not a screen	11
1.29	WINDOWCONTENTS: Grab a window without its borders	12

---

1.30 WINPOS: Tell SGrab where it should open its window . . . . .	12
1.31 MARK: Select area to grab using the mouse . . . . .	12
1.32 VIEWER: Specify an external viewer . . . . .	13
1.33 CLIPVIEWER: Specify an external viewer that can read from clipboard . . . . .	13
1.34 WAITREFRESH: Wait some ticks to let the OS refresh the screen . . . . .	13
1.35 AREXXPORT: Specify name of arexx port . . . . .	13
1.36 NOBLIT: Don't use the blitter to grab screens <= 8bit . . . . .	14
1.37 DATATYPE: Select datatype that should be used for saving . . . . .	14
1.38 CHECKDTMWRITE: List only datatypes that support writing . . . . .	14
1.39 JPG: save as jpeg . . . . .	14
1.40 JPG_QUALITY: Set jpeg quality . . . . .	15
1.41 JPG_SMOOTHING: Set jpeg smoothing factor . . . . .	15
1.42 JPG_PROGRESSIVE: Create progressive jpeg image . . . . .	16
1.43 , . . . . .	16
1.44 , . . . . .	16
1.45 , . . . . .	17
1.46 Graphical User Interface . . . . .	17
1.47 The ARexx Interface . . . . .	18
1.48 Differences between SGrab and CyberGrab . . . . .	20
1.49 How did it grow up . . . . .	20
1.50 How it will grow up . . . . .	21
1.51 Known bugs . . . . .	21
1.52 Acknowledgments . . . . .	21
1.53 Who did it . . . . .	22
1.54 Pick and click . . . . .	22

---

# Chapter 1

## SGrab.doc

### 1.1 Table of Contents

SGrab 1.13

A screen and window grabber that is based on CyberGrab by Stefan Proels.

Purpose

Legal stuff

Requirements

Installation

Configuration and Usage

Graphical user interface

The arexx interface

SGrab vs. CyberGrab

History

Future

Known bugs

Thanks

Author

Copyright (c) 1998-2000 by Stephan Rupprecht

### 1.2 What to do with it

---

## Purpose

~~~~~

SGrab is a utility which allows you to save screens or windows as ILBM or JPEG files.  
SGrab supports all display-modes provided by p96/cgfx or by the native chipsets.

This program is based on Stefan Proels' great CyberGrab. I decided to write a new grabber, because Stefan lost the source code of CyberGrab so that he isn't able to make changes to it :(

## 1.3 Legal stuff

### Copyright

~~~~~

SGrab is Copyright (c) 1998-2000 by the author Stephan Rupprecht

SGrab is distributed under the concept of GIFTWARE, you are requested to send me a small gift if you like the program.

You are allowed to redistribute verbatim copies of this archive as long as all parts stay complete and unmodified. Unpacking or repacking using another archiver is not considered to be a modification as long as all files, including their names, are kept intact.

SGrab must not be used for military purposes. This includes installation on machines used for military purposes.

This program uses ChunkyToPlanar.asm by Morten Eriksen.  
All mentioned trademarks are subject to their owners.

## 1.4 What you need to use it

### Requirements

~~~~~

- An Amiga, CPU 68020 or better
- OS 3.0 or better
- notifyintuition.library (optional) for automatic refresh of the "Screens & Windows" list.
- easyrexx.library (optional) for the ARexx interface.
- datatypes.library v45 (optional).

## 1.5 How to install it

### Installation

~~~~~

Just copy the supplied files to any place you want.

If you want SGrab to talk, ahm, write to you in another language

---

than english, copy the appropriate catalog to

```
LOCALE:Catalogs/whatever/
```

or simply copy the complete Catalogs directory to SGrab's home directory so that it can find the catalog in its PROGDIR:Catalogs/...

If you've just unpacked the original archive and do not intent to move the executable to another directory, SGrab is already prepared to find its Catalog.

Currently there is a german, italian, norwegian, french, spanish and swedish translation.

## 1.6 How to configure and use it

### Configuration and Usage

~~~~~

You can run SGrab from Workbench or from a shell. For WB usage you'll have to specify options via tooltypes, otherwise you'll have to use commandline parameters.

After you've started SGrab it'll wait a little letting you arrange what you want to grab (see below). When the time has passed it'll lock the screen and grab its image data. Finally the picture will be saved as normal ILBM or ILBM24, depending on the screen's depth.

Starting version 1.4 of SGrab a graphical user interface is also available.

Note that grabbing a big screen can take some seconds, so don't bother when graphical operations on that screen are frozen meanwhile.

The following options are currently supported:

```
FILE
  Specify a filename

ICON
  Define an icon template

CX_POPKEY
  Hotkey that tells SGrab to open its window

CX_POPUP
  Open user interface when SGrab gets started

CX_PRIORITY
  Priority of commodity handler

GRABSCRKEY
  Hotkey to grab the current screen
```

---

GRABWINKEY  
Hotkey to grab the current window

PUBSCREEN  
Name of pubscreen to open window on

CLIPUNIT  
Save to clipboard

NUMBER  
Number files all the way through

DELAY  
Define time to wait before grabbing

WINDOW  
Grab a window, not a screen

WINDOWCONTENTS  
Grab a window without its borders

WINPOS  
Tell SGrab where it should open its window

MARK  
Select area to grab using the mouse

VIEWER  
Specify an external viewer

CLIPVIEWER  
Specify an external viewer that can read from clipboard

WAITREFRESH  
Wait some ticks to let the OS refresh the screen

AREXEXPORT  
Specify name of arexx port

NOBLIT  
Do not use the blitter to grab screens <= 8bit

CHECKDTMWRITE  
List only datatypes that support writing

NOBEEP  
Do not flash the screen after grabbing

DATATYPE  
Select datatype that should be used for saving

JPG  
Save as jpeg

JPG\_QUALITY  
Set jpg quality

---



JPG\_SMOOTHING  
Set jpg smoothing factor

JPG\_DCT  
Set jpg dct method

JPG\_PROGRESSIVE  
Create progressive jpeg images

X,Y,W,H  
Specify area to grab by its coordinates

## 1.7 FILE: Specify a filename

FILE

~~~~

The name of file to write the ILBM to. It should include an absolute path, otherwise the file will be saved in SGrab's working directory (i.e. your current working directory if ran from a shell).

The specified filename is not necessarily just a simple identifier representing exactly one disk object. Instead, it is a template which may contain keywords which will be replaced by their actual values whenever a screen or window is grabbed. Keywords are surrounded by braces, they are not case sensitive. For example

```
FILE=sys:gfx/pic/cg_{size}_{number}.ilbm
```

might become

```
sys:gfx/pic/cg_1024x768x8_042.ilbm
```

You can also think of "keywords" as "variables" or "macros", but as these kind of objects usually have characteristics (like being user definable) which are not true for SGrab's keywords I prefer to refer to them as "keywords".

If the actual filename created this way should contain characters which are not valid for AmigaDOS filenames or which are conflicting with AmigaDOS pattern, those characters will be replaced by '\_'s. Note that (especially when using keywords such as {title}) filenames may become too long for AmigaDOS. SGrab will truncate such filenames. Special care will be taken for the keyword {number} (see below). If the number of a file should "jut out" it will be shifted to the left forcing it to fit in.

However, you might not be happy with shortened filenames as they tend to look ugly and an IYHO important part might be cut off. Thus, consider which parts of the filename you need or just would like to have and arrange the keywords as appropriate.

Another way to shorten a filename is to limit the number of words a keyword can be expanded to. This is done by specifying

```
{n:keyword}
```

where n is the maximum number of words which will replace the keyword. For example {2:title} means "take the first two words of the title". Note that this notation is allowed for any keyword, however, (unless n equals 0) it will only affect some of them (because not all consist of multiple words), namely these are {title} and {owner}. Also note that SGrab might have a different idea of what a word is than what you might have expected... The only thing we all surely agree to accept as word separator is a blank -- however this won't be enough for practical cases (consider "foo:bar/baz.stuff" as title of an editor's Window, for example). Thus the definition of the set of word separators is necessarily based on guesses.

The following keywords are currently supported

NUMBER

WIDTH

HEIGHT

DEPTH

SIZE

TITLE

OWNER

TYPE

TIME

DATE

See also:

NUMBER

## 1.8 NUMBER: Number files all the way through

{number} will be replaced by the next greatest number if another file with (except for the number) the same filename does already exist.

Say, for example, you specified `FILE=sys:gfx/cg_{number}.ilbm` and `'list sys:gfx'` shows

```
cg_001.ilbm
cg_030.ilbm
cg_041.ilbm
```

Then the created file will be named `sys:gfx/cg_042.ilbm`.

Note that SGrab needs a way to determine the end of a number in an actual filename, thus you should not put digits (or keywords which are replaced by a sequence of digits) directly after `{number}`. For example `FILE=sys:gfx/cg_{number}{depth}.ilbm` would not have been a good idea in the example above. SGrab will always interpret the largest sequence of digits found in an actual filename (starting at `{number}` in the filename template) as the number of the file.

A number must be unique, thus, it is not possible to put multiple `{number}s` in the filename template.

See also: Option  
NUMBER

## 1.9 WIDTH: The width of a screen of window

`{width}` will be replaced by the width of the screen or window in pixels.

See also:  
HEIGHT  
DEPTH  
SIZE

## 1.10 HEIGHT: The height of a screen of window

`{height}` will be replaced by the height of the screen or window in pixels.

See also:  
WIDTH  
DEPTH  
SIZE

## 1.11 DEPTH: The depth of a screen

`{depth}` will be replaced by the depth of the screen.

See also:  
WIDTH

---

HEIGHT

SIZE

## 1.12 SIZE The size of a screen of window

{size} will be replaced by the size of the screen or window in pixels. This is exactly the same as {width}x{height}x{depth}.

See also:

WIDTH

HEIGHT

DEPTH

## 1.13 TITLE: The title of a screen of window

{title} will be replaced by the title of the screen or window. If the object in question is untitled it will be named "unnamed".

See also:

OWNER

## 1.14 OWNER: The owner of a screen of window

{owner} will be replaced by the owner of the screen or window.

Note: It is not always possible to determine the owner of a screen or window.

If the window is just used for output (i.e. does not have gadgets, a Close-Button, etc.) there is no way to figure out who has opened the window. Technically speaking: A window needs to have a UserPort to be able to determine its owner.

A screen does actually never have owner. Thus {owner} might not be particularly useful when grabbing screens. SGrab will however try to find a somehow sensible name for {owner} even when used with a screen: It will look for a window on the screen and refer its owner as owner of the screen.

If the owner cannot be determined {owner} will become "unknown".

See also:

TITLE

---

## 1.15 TYPE:

{type} will be replaced by the filetype. If you don't use a datatype for saving, this will be ILBM. Otherwise the id of the datatype.

## 1.16 TIME:

{time} will be replaced by the current time (HHhMMmSSs).

## 1.17 DATE:

{date} will be replaced by the current date (dd-mmm-yy).

## 1.18 ICON: Define an icon template

ICON/K

~~~~~

If you want an icon to be created for the FILE you can specify a template-icon using this parameter. This icon can contain tooltypes and a default tool you want to be invoked when you click on the icon. A sample icon named "Template" is supplied with this package. By default SGrab doesn't create icons.

See also:

FILE

## 1.19 CX\_POPKEY: Hotkey that tells SGrab to open its window

CX\_POPKEY/K

~~~~~

This is the hotkey, the input sequence which opens the window if it is closed or brings it to the front if it is already open. You MUST specify this option if you want to use the

gui

.

Example (thanks to by Raymond L. Zarling)

CX\_POPKEY="-capslock shift ctrl numericpad \*\*"

-> Pressing "shift ctrl PrtScr" will open the window.

## 1.20 CX\_POPUP: Open user interface when sgrab gets started

```
                CX_POPUP/T
~~~~~
This one tells SGrab to open its window when it is run for the first time. You
must specify
                CX_POPKEY
to make this option work.
```

## 1.21 CX\_PRIORITY: Priority of commodity handler

```
                CX_PRIORITY/K/N
~~~~~
Priority of commodity handler. Normally, you won't have to change this.
```

## 1.22 GRABSCRKEY: Hotkey that grabs the current screen

```
                GRABSCRKEY/K
~~~~~
This is the hotkey, the input sequence which allows to grab the current
screen.

Example (thanks to by Raymond L. Zarling)
GRABSCRKEY="-capslock shift numericpad **"
-> Pressing "shift PrtScr" will grab the current screen.
```

## 1.23 GRABWINKEY: Hotkey that grabs the current window

```
                GRABWINKEY/K
~~~~~
This is the hotkey, the input sequence which allows to grab the current
window.

Example (thanks to by Raymond L. Zarling)
GRABWINKEY="-capslock shift alt numericpad **"
-> Pressing "shift alt PrtScr" will grab the current window.
```

## 1.24 PUBSCREEN: Screen to open the window on

```
                PUBSCREEN/K
~~~~~
Name of public screen to open window on. If not specified the default
public screen is used.
```

---

## 1.25 CLIPUNIT: Clipboard unit to save image data to

CLIPUNIT/N/K

~~~~~

If you specify CLIPUNIT SGrab will save the screenshot to the given clipboard unit (eg. CLIPUNIT=0). NOTE: datatypes.lib always saves to to the primary unit.

## 1.26 NUMBER: Number files all the way through

NUMBER/S

~~~~~

Usually SGrab will save the picture in the file specified by the FILE option, thereby probably overwriting an existing file. With this option however, generated files will be numbered all the way through. I.e. the file actually created will be named <FILE>.<n> where <n> is the next greatest number of any already existing files matching "<FILE>.[0-9]".

This option is ignored if you do not specify the FILE option in which case a file-requester will be opened.

See the docs to the keywords in the FILE option for a more flexible way to number files.

See also:

FILE

## 1.27 DELAY: Define time to wait before grabbing

DELAY/K/N

~~~~~

Number of seconds to wait before start grabbing.  
Default: 5.

## 1.28 WINDOW: Grab a window, not a screen

WINDOW/S

~~~~~

Usually SGrab grabs the frontmost screen. If you specify this option the currently activated window will be grabbed instead.

NOTE: The operating system provides several types of windows (refresh-types, to be more precise) and not all of them can be grabbed if they are obscured. Actually, only a few can. Thus, you should make sure that the window you wanna grab is completely visible. If a part of a window cannot be grabbed because it is hidden it'll most probably be colored in background color or be garbled. An

easy way to ensure "grabability" is to move the window in front of all other windows before running SGrab.

ALSO NOTE: An applications can prevent SGrab from reading its window. Only very few do, though. Actually, I don't know any :-). If an application installs custom ClipRects you'll notice the same behavior as if a part of the window was obscured.

See also:

WINDOWCONTENTS

## 1.29 WINDOWCONTENTS: Grab a window without its borders

WINDOWCONTENTS/S

~~~~~

This option implies WINDOW but does not grab the window's borders.

NOTE: The operating system provides special windows, called "GimmeZeroZero" which "hide" their borders. SGrab can detect such windows and will always use WINDOWCONTENTS for them, even if you don't explicitly specify this option.

See also:

WINDOW

## 1.30 WINPOS: Tell SGrab where it should open its window

WINPOS/K

~~~~~

This option allows you tell SGrab where it should open its window on a popup. The following keywords are recognized:

MOUSE: opens the window relative to the mouse pointer.

CENTER: centers the window on the screen.

TOP: opens the window at the left edge of the screen right below the title bar.

If you don't specify WINPOS, SGrab will remember the last window position.

## 1.31 MARK: Select area to grab using the mouse

MARK/S

~~~~~

Allows you to select which part of the screen should be grabbed. When SGrab changes the mouse-pointer to a cross, move the mouse pointer to the starting point, hold down the lmb, move to the ending point and release the button. Pressing the rmb will abort this operation.

---



### 1.32 VIEWER: Specify an external viewer

```

VIEWER/K
~~~~~
Viewer to use when you click on the "View" gadget. See also
GUI
.
Default: SYS:Utilities/MultiView "%s"
The %s will be replaced by the filename. IMPORTANT: If you specify your
own viewer you should use *"%s*" instead of %s!

```

### 1.33 CLIPVIEWER: Specify an external viewer that can read from clipboard

```

CLIPVIEWER/K
~~~~~
Same as
VIEWER
, but this one should be able to read from clipboard.
See also
GUI
. Default: SYS:Utilities/MultiView CLIPBOARD CLIPUNIT %ld
The %ld will be replaced by the clipunit specified via
CLIPUNIT
.

```

### 1.34 WAITREFRESH: Wait some ticks to let the OS refresh the screen

```

WAITREFRESH/N/K
~~~~~
This option only takes effect when you use the
GUI
!
SGrab automatically closes its gui when you grab the current screen. In
order to allow the system to refresh the underlying parts of the screen,
SGrab waits some ticks (1 tick = 1/50 sec) before it starts grabbing.
WAITREFRESH can be used to change the size of this time gap, eg. if you
have a slow system you may wish to enlarge it. Default: 25 ticks.

```

### 1.35 AREXSPORT: Specify name of arexx port

```

AREXSPORT/K
~~~~~
Allows you to change the name of the arexxport. Default: SGRAB
See also
Arexx commands

```

### 1.36 NOBLIT: Don't use the blitter to grab screens <= 8bit

NOBLIT/S

~~~~~

This option has been introduced to fix a deadlock problem when MCX is used. Setting this option avoids the deadlock but slows down the grabbing process of planar screens. With a gfx-board you may gain a little speed when this option is used.

### 1.37 DATATYPE: Select datatype that should be used for saving

DATATYPE/K

~~~~~

This option allows you to specify a datatype that should be used for saving. By default the internal ILBM encoder is used. DATATYPE also changes the initial state of the filetype gadget

NOTE: You can only save ilbm pictures to the clipboard. This option requires datatypes.library v45.

### 1.38 CHECKDTMWRITE: List only datatypes that support writing

CHECKDTMWRITE/S

~~~~~

By default the filetype gadget shows all (picture) datatypes that are currently available, no matter if they support writing imagedata in their local format or not. Setting this option forces SGrab to list only those datatypes that support writing, for this purpose all datatypes must be loaded into memory and thus delays popping up the window. NOTE: Requires datatypes.library v45.

### 1.39 JPG: save as jpeg

JPG/K

~~~~~

If you don't specify

---

```

        DATATYPE
        , SGrab will save the imagedata as
IFF-ILBM. When setting JPG the image will be saved as jpeg instead. JPG
also changes the initial state of the
        filetype gadget
        .

```

NOTE: SGrab requires jpeg.library by Paul Huxham (util/libs/jpeglibrary\*.lha). You can find the latest version of this library on Pauls homepage at <http://mafeking.scouts.org.au/~paulh/>.

See also:

```

        JPG_QUALITY
        ,
        JPG_SMOOTHING
        ,
        JPG_PROGRESSIVE
        .

```

## 1.40 JPG\_QUALITY: Set jpeg quality

```

        JPG_QUALITY/N/K

```

```

~~~~~

```

This option allows to change the quality of the jpeg image to create.  
Default: 85 (Range: 1% - 100%)

See also:

```

        JPG
        ,
        JPG_SMOOTHING
        ,
        JPG_DCT
        ,
        JPG_PROGRESSIVE
        .

```

## 1.41 JPG\_SMOOTHING: Set jpeg smoothing factor

```

        JPG_SMOOTHING/N/K

```

```

~~~~~

```

This option allows to change the smoothing factor of the jpeg image to create. Default: 0 = no smoothing (Range: 0% - 100%)

See also:

```

        JPG
        ,
        JPG_QUALITY
        ,
        JPG_DCT
        ,
        JPG_PROGRESSIVE

```

---

## 1.42 JPG\_PROGRESSIVE: Create progressive jpeg image

```

                JPG_PROGRESSIVE/S
~~~~~
Setting this option will cause jpeg.library (v5 is required) to create
progressive images.

```

See also:

```

                JPG
                '
                JPG_QUALITY
                '
                JPG_DCT
                '
                JPG_SMOOTHING
                .

```

## 1.43 ,

```

                JPG_DCT/K
~~~~~
JPG_DCT allows you to use a different dct method for saving. The following
methods are available:

```

```

ISLOW: slow but good quality (default).
IFAST: less accurate but fast.
FLOAT: best quality but slow (depending on your cpu/fpu).

```

NOTE: This option requires version 3 or higher of jpeg.library!

See also:

```

                JPG
                '
                JPG_QUALITY
                '
                JPG_SMOOTHING
                '
                JPG_PROGRESSIVE
                .

```

## 1.44 ,

```

                X, Y, W, H/N/K
~~~~~
These four options determine the area that should be grabbed. X stands

```

---

for the x starting point, Y for the y one (default 0). W and H stand for the width and height of the area (default screen width/height).

NOTE: These options are ignored when grabbing a window or using  
MARK

.

## 1.45 ,

NOBEEP/S

~~~~~

Turns off screen-flashing after the grab.

## 1.46 Graphical User Interface

HOW IT WORKS

~~~~~

Whenever you start SGrab with

CX\_POPKEY

given, SGrab will install itself

as a commodity into the system. Each time you press the specified input sequence, SGrab will open its GUI on the screen specified by the

PUBSCREEN

option (screen and window will be brought to front if already ←  
open).

THE GADGETS

~~~~~

Screens / Windows: This one shows all windows and screens currently available. The highlighted entries are the screens. You can use the CRSR UP/DN keys to select an entry via the keyboard. If you have notifyintuition.library by H.W.Schober installed the gadget will be automatically updated each time when a window/screen is opened or closed. For windows that don't have a title, their coordinates and dimensions are shown instead.

Windows: Turns on / off the displaying of windows in the "Screens / Windows" gadget. The initial state of this gadget can be controlled by the

WINDOW

option.

WINDOWCONTENTS: See

WINDOWCONTENTS

.

Grab: Grabs the selected screen / window.

Mark & Grab: Pops the selected screen to the front and allows you to select the area to grab. See also

MARK

.

View: Views the last grab by using an external viewer. See also  
VIEWER

,

CLIPVIEWER

.

slider gadget: Allows you to specify the time (in seconds) before SGrab actually starts a grab operation. See also

DELAY

.

You can use CRSR left / right to control this gadget via the keyboard.

File: Where to save the imagedata to. See also

FILE

.

Clipboard: Save imagedata to the clipboard unit specified by

CLIPUNIT

(0 by default).

Filetype: Here you can select the datatype that should be used to save the imagedata. If you select 'ILBM (internal)', SGrabs' internal saver will be used. Using the internal saver is fastest way to grab a screen and requires less memory than datatypes. NOTE: Only a few datatypes support saving.

See also:

DATATYPE

,

CHECKDTMWRITE

.

The following two gadgets are only shown when jpeg.library is installed and 'jpeg (internal)' of Filetypes gadget is chosen.

JPEG Quality: See

JPG\_QUALITY

.

JPEG Smoothing: See

JPG\_SMOOTHING

.

## THE MENU

~~~~~

The menu only contains the standard menu items "About", "Hide" and "Quit" so I don't think I have to explain them to you :)

## 1.47 The ARexx Interface

ARexx

~~~~~

To use the arexx interface you must have easyrexx.library by Ketil Hunn installed (dev/misc/easyrexx.lha). Currently the following commands are

available (most of them are similar to the wb/shell options):

```
GRABSCREEN FILE,CLIPBOARD/S,SCREEN/K,DELAY/N/K,DESTDT=DATATYPE/K,JPG/S,
           Q=JPG_QUALITY/N/K,S=JPG_SMOOTHING/N/K,MARK/S,X=XSTART/N/K,
           Y=YSTART/N/K,W=WIDTH/N/K,H=HEIGHT/N/K,P=JPG_PROGRESSIVE/S
GRABWINDOW FILE,CLIPBOARD/S,WINDOW/K,DELAY/N/K,WINDOWCONTENTS=WINC/S,
           DESTDT=DATATYPE/K,JPG/S,Q=JPG_QUALITY/N/K,S=JPG_SMOOTHING/N/K,
           P=JPG_PROGRESSIVE/S
```

FILE: See

FILE

.

CLIPBOARD: save to clipboard.

SCREEN: Screen that should be grabbed identified by its title. You don't need to give the whole title. If you don't specify SCREEN the current screen will be grabbed.

WINDOW: Grab the window identified by its title. See also SCREEN.

WINDOWCONTENTS: Don't grab the window borders. See also

WINDOWCONTENTS

.

DATATYPE: See

DATATYPE

.

DELAY: See

DELAY

.

JPG: See

JPG

.

JPG\_QUALITY: See

JPG\_QUALITY

.

JPG\_SMOOTHING: See

JPG\_SMOOTHING

.

JPG\_PROGRESSIVE: See

JPG\_PROGRESSIVE

.

MARK: See

MARK

.

X,Y,W,H: See cli opts

X,Y,W,H

.

QUITSGRAB

Guess what this option does ;)

The arexxport is named SGRAB and can be changed by the

AREXXPORT

option.

NOTE: You need to invoke SGrab with

CX\_POPKEY

given in order to use

the interface.

## 1.48 Differences between SGrab and CyberGrab

Differences

~~~~~

```
\textdegree{}~sgrab has a GUI
\textdegree{}~option SPAREMEM not supported - does anybody need it?
\textdegree{} option WINDOWFRAME not available, SGrab automatically uses
WINDOWFRAME whenever possible.
\textdegree{}~grabs planar screens faster
\textdegree{} clipboard support
\textdegree{} datatypes can be used
\textdegree{} ...
```

## 1.49 How did it grow up

History

~~~~~

- 1.0 first public release
- 1.1 crashed when grabbing large screen (compiler used 16bit math instead of 32bit ones).
- 1.2 italian catalog by Nicola Pagani, norwegian catalog by Kimme Utsi, fixed stripes problem when grabbing < 8bit cgfx screens (only when SAVEMEM wasn't activated) <juergen urbanek>, added commodity support <freejack>, supports clipboard now
- 1.3 for unknown reasons grabbing didn't work correctly on at least one computer (MARK and WINDOW) when using a friend bitmap (friend bitmaps were only used for native screens) <Henning 'Faroul' Peters>, new mouse pointer image when using MARK
- 1.4 finally sgrab got a gui, wbmsg wasn't replied, sometimes crashed the system when {Owner} / {Title} were used, flashes the screen when grabbing has been done.
- 1.5 {NUMBER} was broken when using the GUI. Reopens the window when it was closed before a grab. <Raphael Pilarczyk> french catalog by Didier Giron
- 1.6 fixed a bug (not a real one, just a side-effect when using MCP or something similar) where the MARK operation seemed to "hang". Added a function to start an external viewer. DELAY is available from the gui now. MARK can be aborted by pressing the rmb. Some minor improvements. Finally, with help of Raphael Pilarczyk I found out why SGrab deadlocked Raphaels' machine, still not fixed :( See "Known bugs" for more.
- 1.7b missing swedish catalog by F. Söderberg is now available. By request of a SINGLE user I implemented an arexx interface. New NOBLIT option fixes problems with MCX and speeds up grabbing of <= 8bit screens when a gfxboard is installed. datatypes can be used for saving image data.
- 1.7 Added support for jpeg.library by Paul Huxham. Removed arexx command MARKGRAB, GRABSCREEN has a MARK option now. Some internal changes and bugfixes.



- 1.8 French catalog has been updated. AmigaGuide® doc could crash the system <Didier Giron>. Now you can change the dct method used when saving jpeg images (requires version 3+ of jpeg.library). new options to specify a area to grab by its coordinates. MARK function will display the current coordinates now.
- 1.9 fixed p96 picturedt related problems. improved chunky2planar routine. added support for jpeg.library v5 (can create progressive images). NOBEEP option turns off screen flashing after grabbing. two new options to grab the current window/screen by pressing a hotkey. new keywords: TIME and DATE (see FILE option).
- 1.10 jpeg saver supports ham and ehb now.
- 1.10a revised guide. included new .guide icons by Pavel Narozny
- 1.11 enforcer hit when reading current window list <Nils Goers>, <Vit Sindlar>, <Allan Odgaard>. ByteRunl compression could cause crashes. Pressing the hotkey while the window is open no longer brings it to front but closes it instead <Allan Odgaard>.
- 1.12 Olaf Barthel found a bug in the original ByteRunl compressor, SGrab uses Olafs' update now. when using 'mark & grab', the width and height was off by two, when marking from right to left or bottom to top <Allan Odgaard>.
- 1.13 added WINPOS/K option. the {date} keyword used substitutions (Today, ← Monday, Yesterday, ...) instead of the date in numerical format <Olivier Fabre>.

## 1.50 How it will grow up

Future

~~~~~

```
\textdegree{} new gui (using MUI, RA or something similiar)
\textdegree{} png saver
\textdegree{} bugfixes ;)
\textdegree{} your suggestions
```

## 1.51 Known bugs

Bugs

~~~~

MCX: The titlelock of this program does not lock the barlayer before it writes to the titlebar. This leads to deadlocks when SGrab tries to lock the screen. The  
 NOBLIT  
 Option of SGrab may help to avoid this  
 problem.

jpeg.library: The ffp version of this library may cause your computer to crash. Better use the standard or fpu one.

## 1.52 Acknowledgments

### Thanks

~~~~~

Special thanks go to Stefan Proels for allowing me to use the CyberGrab documentation for SGrab.

Dirk Neubauer for beta-testing on p96.  
Dhomas Trenn for the bug report.  
Nicola Pagani for the italian catalog.  
Kimme Utsi for the norwegian catalog.  
Arturo Roa for the spanish catalog.  
Juergen Urbanek for the bug report.  
Didier Giron for the french catalog and correcting the guide of v1.7 (which crashed his system).  
Raphael Pilarczyk for his bug report concerning MCX.  
Fredrik Söderberg for his suggestions and the swedish catalog.  
Zap for his bug report.  
Greg Reyna for his suggestions and testing pre-releases.  
Pavel Narozny for the icons and suggestions.  
Jon B. Peterson for the gift.  
Thomas Franziskowski for the gift.  
... see

history  
:)

## 1.53 Who did it

Author

~~~~~

Send comments, questions, etc., to

stephan.rupprecht@gmx.de

You may also contact me via snailmail:

Stephan Rupprecht  
Apfeldweg 1  
44359 Dortmund  
Germany

WWW

<http://home.debitel.net/user/stephan.rupprecht>

## 1.54 Pick and click

General

Acknowledgments

---

ARexx

Author

Configuration

Copyright

GUI

History

Future

Installation

Known bugs

Legal Stuff

Overview

Purpose

Requirements

Table of Contents

Thanks

Options

FILE

Keywords in FILE argument

DEPTH

HEIGHT

NUMBER

OWNER

SIZE

TITLE

WIDTH

TYPE

TIME

DATE

DELAY

---

ICON  
CX\_POPKEY  
CX\_POPUP  
CX\_PRIORITY  
GRABSCRKEY  
GRABWINKEY  
PUBSCREEN  
CLIPUNIT  
NUMBER  
WINDOW  
WINDOWCONTENTS  
WINPOS  
MARK  
VIEWER  
CLIPVIEWER  
WAITREFRESH  
DATATYPE  
AREXEXPORT

---