

**CyberGrab.doc**

**COLLABORATORS**

	<i>TITLE :</i> CyberGrab.doc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 20, 2024	

**REVISION HISTORY**

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

# Contents

<b>1</b>	<b>CyberGrab.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	3
1.8	NUMBER: Number files all the way through	5
1.9	WIDTH: The width of a screen of window	5
1.10	HEIGHT: The height of a screen of window	5
1.11	DEPTH: The depth of a screen	5
1.12	SIZE: The size of a screen of window	6
1.13	TITLE: The title of a screen of window	6
1.14	OWNER: The owner of a screen of window	6
1.15	ICON: Define an icon template	6
1.16	NUMBER: Number files all the way through	7
1.17	DELAY: Define time to wait before grabbing	7
1.18	WINDOW: Grab a window, not a screen	7
1.19	WINDOWCONTENTS: Grab a window without its borders	8
1.20	WINDOWFRAME: Grab a part of a screen framed by a window	8
1.21	SPAREMEM: Use less memory	8
1.22	What's new	9
1.23	How did it grow up	9
1.24	Acknowledgments	10
1.25	Who did it	10
1.26	Pick and click	11

---

# Chapter 1

## CyberGrab.doc

### 1.1 Table of Contents

CyberGrab 1.2

A screen and window grabber for the CyberGraphX RTG system

Purpose

Legal stuff

Requirements

Installation

Configuration and Usage

Updating from V1.1

History

Thanks

Author

Copyright (c) 1995 by Stefan Pröls

### 1.2 What to do with it

Purpose

~~~~~

If you can answer one of the following questions with "YES"  
you won't need this program:

- You have no Gfx-Board
- You're not running CyberGraphX
- You're not missing some kind of Screen-Grabber

So, you're still here? Ah, you've answered all questions with "NO", ok,  
then read on...

---

CyberGrab is a utility which allows you to save Screens or Windows as "normal" ILBM or ILBM24 files.

CyberGrab supports all Display-Modes provided by the CyberGraphX system, including 8, 15, 16 and 24 bit modes.

### 1.3 Legal stuff

Copyright

~~~~~

CyberGrab is Copyright (c) 1995 by the author Stefan Proels.

CyberGrab is distributed under the concept of Freeware, this means that you may 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.

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

### 1.4 What you need to use it

Requirements

~~~~~

- An Amiga, CPU 68020 or better
- OS 3.0 or better
- CyberGraphX 40.46 or better

Note: As CyberGraphX requires at least a 020er this program has been compiled for this processor too.

Also note: Unless you're using screens deeper than 8 bit CyberGrab should work without CyberGraphX, i.e. can be used with native Amigas too. This has however never been tested and is not a recommended usage as it will be quite slow because of redundant chunky<->planar conversions.

The amount of memory required depends heavily on the screen (or window) you want to grab. For example, for an 800x600x8 image about 500kB will be sufficient, while for an 1024x768x24 image you'll need about 2.5MB. Since version 1.2 CyberGrab may do with much less memory (at the expense of speed) if it cannot to allocated the desired amount.

### 1.5 How to install it

Installation

~~~~~

Just copy the supplied files to any place you want.

If you want CyberGrab 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 CyberGrab'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, CyberGrab is already prepared to find its Catalog.

Currently there is only a german translation.

## 1.6 How to configure and use it

### Configuration and Usage

~~~~~

You can run CyberGrab 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 CyberGrab 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.

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                    |
| 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          |
| WINDOWFRAME    | Grab a part of a screen framed by a window |
| SPAREMEM       | Use less memory                            |

## 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 CyberGrab's working directory (i.e. your current working directory if ran from a shell).

If no filename can be determined a file-requester will be opened letting you choose a filename.

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 CyberGrab'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. CyberGrab 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 CyberGrab 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
```

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 CyberGrab 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. CyberGrab 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. CyberGrab 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 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 CyberGrab doesn't create icons.

See also: FILE

## 1.16 NUMBER: Number files all the way through

NUMBER/S

~~~~~

Usually CyberGrab 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

## 1.17 DELAY: Define time to wait before grabbing

DELAY/K/N

~~~~~

Number of seconds to wait before start grabbing.  
Default: 3.

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

WINDOW/S

~~~~~

Usually CyberGrab 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 CyberGrab.

ALSO NOTE: An applications can prevent CyberGrab 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

## 1.19 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. CyberGrab can detect such windows and will always use WINDOWCONTENTS for them, even if you don't explicitly specify this option.

See also: WINDOW

## 1.20 WINDOWFRAME: Grab a part of a screen framed by a window

WINDOWFRAME/S

~~~~~

This option implies WINDOW but does not actually grab the image of a window but the part of a screen which is occupied by a window. Thus, WINDOWFRAME is a somehow hybrid option -- not really WINDOW but also not not WINDOW :-)

This is useful, for example, if have a big window on a screen with several other windows on it and want to grab what you see in the big window but not the whole screen.

You can also use WINDOWFRAME as "trick" to grab a e.g. SuperBitMap window including its borders (see the docs to the WINDOWCONTENTS option for an explanation why WINDOW can't do that).

See also: WINDOW WINDOWCONTENTS

## 1.21 SPAREMEM: Use less memory

SPAREMEM/S

~~~~~

Try to reduce the amount of memory used for grabbing.

In the worst case (i.e. if image compression has no effect) this option can't reduce memory requirements either, however, in the average case this option will save quite a lot of memory.

It works as follows: Usually CyberGrab will compute the maximum amount of memory required to store the complete image and allocate one chunk of memory of that size. With SPAREMEM enabled it will use smaller buckets of memory just large enough to store a couple of

lines, i.e. it will grab some lines, compress them, and copy the data into a smaller bucket just large enough for the compressed couple of lines. Thus, instead of using a huge (usually much larger than actually required) piece of memory it will use a bunch of (usually much smaller) pieces. This can reduce memory usage significantly -- however, few things come without a price: SPAREMEM will slow down the program. It depends on your machine if you'll notice the slow-down -- it should not be dramatically, though (CyberGrab 1.2 with SPAREMEM enabled will still be faster than CyberGrab 1.1 which didn't have this option).

Usually you do not need to specify this option as it will automatically be enabled if the "memory hog" method cannot be used due to the lack of memory. There is however a slight chance that you'll run out of memory AFTER the image has been grabbed using the default method (when opening a file-requester, for example). To make the short story long: If you want to try this option, here it is.

## 1.22 What's new

Updating from version 1.1

~~~~~

This section is dedicated to users of the last release of CyberGrab who do not want to read the whole documentation again.

So, what's new?

First there are some internal enhancements, mostly noticeable in a significant speedup.

Locale support has been added. See Installation for a description on how to make use of it.

The FILE option has been extended. Now it takes a template as argument, providing a more flexible way to define a filename. Please read FILE for a more detailed description.

Finally, there are two new options:

|             |  |
|-------------|--|
| WINDOWFRAME | Grab a part of a screen framed by a window |
| SPAREMEM    | Use less memory                            |

See also: History

## 1.23 How did it grow up

History

~~~~~

1.0 first pre-release

1.1 first public release

---

- \* fixed a spelling bug in the program's name |-)
  - \* added image compression
  - \* changed grabbing technique, much faster now but screen gets locked longer. Fixed an alignment bug on-the-fly.
  - \* added WINDOW and WINDOWCONTENTS options
  - \* added ICON option
  - \* added NUMBER option
  - \* FILE option is no longer required.  
If not supplied a requester is opened.
- 1.2
- \* grabbing much faster  
(about twice as fast on <= 8 bit screens)
  - \* in SuperBitMap windows the correct part is now grabbed even if the window's layer has been scrolled
  - \* now comes with an AmigaGuide documentation
  - \* added locale support
  - \* more flexible way to define the FILEname
  - \* added WINDOWFRAME option
  - \* added SPAPEMEM option
  - \* ^C now terminates the program when DELAYed.  
Changed default DELAY to 3 seconds.

## 1.24 Acknowledgments

Thanks

~~~~~

Special thanks go to Frank Mariak and Thomas Sontowski, the authors of CyberGraphX, who implemented RTG for AmigaOS and therefore are the pioneers as well as the front runners on that score.

## 1.25 Who did it

Author

~~~~~

Send comments, questions, etc., to

Internet: proels@fmi.uni-passau.de  
Fidonet: Stefan Proels,2:2494/22.13

---

You may also contact me via snailmail:

Stefan Pröls  
Rudolf-Guby-Str. 1  
94032 Passau  
Germany

## 1.26 Pick and click

### General

Acknowledgments  
Author  
Configuration  
Copyright  
History  
Installation  
Legal Stuff  
Overview  
Purpose  
Requirements  
Table of Contents  
Thanks  
Updating  
Usage

### Options

#### FILE

Keywords in FILE argument

DEPTH  
HEIGHT  
NUMBER  
OWNER  
SIZE  
TITLE  
WIDTH

DELAY  
ICON  
NUMBER  
SPAREMEM  
WINDOW  
WINDOWCONTENTS  
WINDOWFRAME

---