

**PicBoot**

**COLLABORATORS**

	<i>TITLE :</i> PicBoot		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 29, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>PicBoot</b>	<b>1</b>
1.1	PicBoot.guide	1
1.2	PicBoot.guide/Introduction	2
1.3	PicBoot.guide/System requirements	2
1.4	PicBoot.guide/Legal information	2
1.5	PicBoot.guide/Usage	3
1.6	PicBoot.guide/FILES	4
1.7	PicBoot.guide/MODEID	5
1.8	PicBoot.guide/LIST	5
1.9	PicBoot.guide/DETACH	6
1.10	PicBoot.guide/CENTER	6
1.11	PicBoot.guide/DEFAULT	7
1.12	PicBoot.guide/AUTOSCROLL	7
1.13	PicBoot.guide/VIDEOOVERSCAN	8
1.14	PicBoot.guide/Known problems	8
1.15	PicBoot.guide/GetModeID	9
1.16	PicBoot.guide/UnpackILBM	9
1.17	PicBoot.guide/Author contact	10
1.18	PicBoot.guide/Version history	10
1.19	PicBoot.guide/Version 1.00	10
1.20	PicBoot.guide/Version 1.01	11
1.21	PicBoot.guide/Version 1.02	11
1.22	PicBoot.guide/Version 1.03	12
1.23	PicBoot.guide/Version 2.0	13

---

# Chapter 1

## PicBoot

### 1.1 PicBoot.guide

This file describes PicBoot, version 2.0, a program to show an `↔`  
IFF  
ILBM or GIF picture during boot.

Introduction

What is PicBoot?

System requirements

What you need to run the program

Legal information

Legal information and disclaimer

Usage

Basic usage

Known problems

Problems and bugs

GetModeID

What is GetModeID

UnpackILBM

What is UnpackILBM

Author contact

How to reach me

Version history

Version history

---

## 1.2 PicBoot.guide/Introduction

### Introduction

\*\*\*\*\*

Have you removed all output in your 2.0+ startup, and only see a black screen during boot? Wouldn't it be nice to have a picture instead? A picture that disappeared when the Workbench screen opened?

If so, PicBoot is certainly a program for you. What it will do is to read any IFF file containing an ILBM picture - or a GIF file - and show that picture. As soon as the Workbench screen appears (or you press any mouse-button), the picture will go away.

#### Features:

- \* Fast picture unpacking, using highly optimized assembler. The entire picture is read into memory and then unpacked. This applies both to the IFF and the GIF unpacker.
- \* Optional auto-detaching; the picture is loaded as fast as possible, with minimal memory fragmentation.
- \* Random select among any number of pictures.
- \* Force a certain display mode for a picture, even if saved with another (can be selected on a picture by picture basis, if you are using a LIST file (see LIST)).
- \* Optional screen centering (horisontally).

## 1.3 PicBoot.guide/System requirements

### System requirements

\*\*\*\*\*

Apart from OS 2.04+, PicBoot doesn't require any special libraries. The only non-ROM library required is iffparse.library (which normally resides in Libs:).

## 1.4 PicBoot.guide/Legal information

---

## Legal information

\*\*\*\*\*

This program is freeware. You may copy and use this program freely, as long as the following conditions are met:

- \* All files are copied in an unmodified state. If additional information is needed, place it in a separate file. Preferably redistribute in the original archive form (\*.Lha).
- \* The copying is done on a non-commercial and non-profit basis only. A copy fee to cover media costs, postage etc. may be charged. This fee may not exceed the fee to obtain an AmigaLibDisk from Fred Fish.
- \* The copier/spreader is not claiming the Copyright © of this program.

Any exceptions from these restrictions requires written permission from the author, Magnus Holmgren.

## Disclaimer

=====

Magnus Holmgren neither assume nor accept any responsibility for the use or misuse of these programs. He also will not be held liable for damages or any compensation due to loss of profit or any other damages arising out of the use, or inability to use these programs.

Magnus Holmgren will not be liable for any damage arising from the failure of these programs to perform as described, or any destruction of other programs or data residing on a system attempting to run the programs. While he know of no damaging errors, the user of these programs uses it at his or her own risk.

## 1.5 PicBoot.guide/Usage

## Usage

\*\*\*\*\*

To activate PicBoot, add a line to your S:Startup-Sequence, looking something like this:

```
PicBoot Pics:Hi-res/Calvin01.Pic DETACH
```

Or, if you have a list of files in "Work:Text/PicList":

```
PicBoot Work:Text/PicList LIST DETACH
```

This line should be located near the beginning in the S:Startup-Sequence (no point in placing it near the LoadWB command, is it? :), but keep it after SetPatch. PicBoot will only output any text if it fails, so don't re-direct its output. Note however that if you

place PicBoot before any additional monitors are installed, you'll be stuck with the default.monitor for showing your picture. The DEFAULT switch may be of help here.

Make sure no program makes any output in the CLI window, since then the Workbench screen will open with a boring CLI-window instead...

Options:

```

FILES
    Picture(s) to view, or name(s) of listfile(s)

MODEID
    Use this display mode

LIST
    The pictures to view are stored in a listfile

DETACH
    Detach from the CLI when picture is loaded

CENTER
    Should the screen be centered?

DEFAULT
    Force the default monitor to be used?

AUTOSCROLL
    Enable autoscrolling?

VIDEOOVERSCAN
    Use video overscan?

```

## 1.6 PicBoot.guide/FILES

FILES

=====

This is the only required argument. Here you specify the name of the picture you want to view. You may enter several files here, in which case PicBoot will select one of them randomly, and show that one. It may also be the name of an ASCII file containing a filename list if you specified the LIST option (see

```

LIST
).

```

Note: The random-number generator used is quite simple, and is based on the current system time. I don't know if it is a "good" random generator. It seems to be enough random for me at least.. :) If you have an algorithm you think is better, don't be afraid to send it to me. It should be fairly small.. :)

---

## 1.7 PicBoot.guide/MODEID

MODEID

=====

Short form: M

NOTE: This argument is mainly for the more "advanced" user.

This argument should be a decimal number specifying which displaymode to use. It basically replaces the so called CAMG hunk in an ILBM file. Thus, you must select mode with care, or else the picture will look like trash (nothing more serious can happen. I hope! :). When showing GIF files, it will override the internal "best mode" routines (which aren't good at all. But I haven't bothered to add code to make them better.. :).

To make it easier for you to find out which display mode id to use, there is a small program called GetModeID included, which uses the ReqTools or Asl screenmode requester. Simply select the display mode you want. See

GetModeID

.

The mode id will be passed through the same validity checking as a normal so called CAMG chunk, so PicBoot should handle bad values properly (although I haven't tested this much.. :).

## 1.8 PicBoot.guide/LIST

LIST

=====

Short form: L

If this switch is specified, PicBoot will interpret the files in the FILES argument as names of files containing a list of pictures (or rather, argument lines). PicBoot will then randomly select one of the lines in the selected file, and process it almost like a normal argument line. The only difference is that you can't use the DETACH argument. This argument may not be specified in a listfile (no point in doing it anyway).

The filelist file is an ASCII (text) file with a simple layout. On the first line you specify the number of argument lines in the file. This is usually <number of lines in file>-2 (one line is occupied by the count, and the other is the last linefeed). The rest of the file is



simply the argument lines to choose from.

```
4
Work:Pics/Comics/Calvin01.Pic CENTER ON
Work:Pics/Comics/Calvin02.Pic MODEID 137220
Work:Pics/Comics/Calvin03.Gif
Work:Text/MoreCalvins.txt LIST CENTER ON
```

The first file will be centered (regardless of what the command line said).

The second file have a display mode id added to it. This particular id tells PicBoot to use the PAL monitor, with a low-res interlace HAM screen. If no id is specified, PicBoot will use the MODEID specified in any previous "listfile line", the CLI-arguments, or the contents of the CAMG chunk/"best mode" (in that order).

The last example line is another list file, which PicBoot will load and process. Centering will be on as default for all files in this list.

Note that any arguments specified from the CLI, or in any previous listfile, will be taken as the new default value. In the listfile you may alter this default. This does not include the LIST argument (ofcourse). It is always turned off before parsing a line.

Warning: Since you may enter a new listfile within a listfile, you can be caught in an endless loop, constantly changing (maybe to the same) listfile. No checking for this is done. Also, since there is no CLI-window around, you have no chance to send PicBoot any CTRL-c, if PicBoot should happen to listen to this. You have been warned! :)

Note: A line in a listfile may not be more than 510 chars, or it will be truncated when read. This shouldn't cause any problems I think..

## 1.9 PicBoot.guide/DETACH

```
DETACH
=====
```

Short form: D

If this switch is specified, PicBoot will detach from its calling CLI when the picture is fully loaded and displayed. If you specify this option, you shouldn't "Run" PicBoot. This option will reduce memory fragmentation, and will ensure that the picture gets loaded quickly. However, the implementation maybe isn't the most "system-friendly" one (but there aren't that many alternatives), hence the switch. I don't think there will be any problems, but one can never know... :)

## 1.10 PicBoot.guide/CENTER

---

CENTER  
=====

Short form: C

Possible arguments: YES, ON, NO, OFF. Default is NO.

If this switch is on (argument is YES or ON), PicBoot will center the picture. This centering should work fine for most screen modes, but one can never now.. :) If a screen promotor is active, then PicBoot can get it wrong (when the screen is opened in another mode than PicBoot had asked for).

## 1.11 PicBoot.guide/DEFAULT

DEFAULT  
=====

Short form: DEF

Possible arguments: YES, ON, NO, OFF. Default is NO.

If this switch is on (argument is YES or ON), PicBoot will force the picture to use the default.monitor, regardless of what was actually stored in the picture (in the CAMG chunk). This is needed since very early in the startup, default.monitor is the only monitor available (e.g. multiscan.monitor is normally not available). In the future, I might add more types of "forcing" (e.g. force a picture to PAL, NTSC or whatever that might be useful).

This switch also acts on the MODEID parameter, if specified.

## 1.12 PicBoot.guide/AUTOSCROLL

AUTOSCROLL  
=====

Short form: AS

Possible arguments: YES, ON, NO, OFF. Default is NO.

If this switch is on (argument is YES or ON), the OS 2.0+ autoscrolling of screens will be enabled.

Note: During boot, this switch may make the actual display a bit smaller than normally possible. There is nothing I can do about that... :) You can, however, by ensuring that ENV: is properly set up before PicBoot is started. Or you could try the VIDEOOVERSCAN switch (see

```
VIDEOOVERSCAN
).
```

## 1.13 PicBoot.guide/VIDEOOVERSCAN

```
VIDEOOVERSCAN
=====
```

Short form: VO

Possible arguments: YES, ON, NO, OFF. Default is NO.

If this switch is on (argument is YES or ON), the visible size of the opened screen will be as large as the system can handle. Forces AUTOSCROLL to YES.

## 1.14 PicBoot.guide/Known problems

```
Known problems
*****
```

I do not know of any real bugs in PicBoot. However, certain parts of the program may still contain bugs. E.g., pictures that have a mask bitplane (mskHasMask) are supported, but since I only have one (compressed) picture that have a mask, there might be a bug in that code (can't test it properly). Please report any problems!

Currently there is no support for SHAM, PCHG and similar "special" pictures. I'm not sure if this could be implemented in a "clean" way (that would work on future systems etc). These pictures aren't that common, and I have an Amiga with AA-graphics, so... :) Color cycling is currently ignored (I have no need for it).

Interlaced GIF pictures aren't supported, since I don't have any such picture. Besides, the decompression of such pictures would be slower anyway.

PicBoot doesn't remap GIF files in any way. Even if you have ECS, GIF files can still be useful. This is because a GIF file can have from 2 to 256 colors (inclusive). Thus, if you have a program that can save a 32-color picture as a 32-color GIF file, there will be no problem to view it with PicBoot.

The "best mode" routine used in the GIF reader isn't good at all (this includes the ROM function in OS 3.0+! :).

Pictures with more than 8 bitplanes are currently not supported by PicBoot.

---

The centering for (some?) Super72 screens doesn't work. I suspect this is an OS-bug (I know that PicBoot calculates a reasonable offset, which Intuition seems to ignore).

## 1.15 PicBoot.guide/GetModeID

GetModeID

\*\*\*\*\*

GetModeID is a simple program that shows a Asl or ReqTools screenmode requester, whichever is available. The program will then print out the decimal identifier for the selected screenmode, suitable for use together with the

MODEID

parameter. This program can only be used from a Shell. Example usage:

```
PicBoot Island.Gif 'GetModeID'
```

This will first show a screenmode requester (if you have one, that is), and then show the GIF-picture in the selected screenmode.

## 1.16 PicBoot.guide/UnpackILBM

UnpackILBM

\*\*\*\*\*

UnpackILBM is another simple program (at least in theory... :). It will take any IFF ILBM picture and unpack the so called BODY chunk in it (this is the actual image data). This means that e.g. PicBoot will be able to display that image a little faster, at least if loading them from some fast media. Or, if you use PPSHOW/ShowIFF, you could repack the picture with PowerPacker/Xpk, to maximize the compression (as the compression used in IFF ILBM isn't a very efficient one. But on the other hand, it is rather fast). But then PicBoot won't be able to load them.. :) Example usage:

```
UnpackILBM Island.Pic Island.Pic.NoComp
```

```
UnpackILBM Island.Pic
```

The first example will unpack the picture to a new one, while the other will - via a temporary file - overwrite the original picture with the uncompressed version.

I wrote this program since someone (I've forgot who) wanted it... :)

Note: I haven't tested this program that much. I've converted a few pictures, so it seems to work fine (at least when there aren't any errors), but one can never know.. Please report any problems!

Note: This program doesn't strip any information. All chunks will remain. The picture data is only decompressed.

## 1.17 PicBoot.guide/Author contact

Author contact  
\*\*\*\*\*

PicBoot was written in 100% assembler by Magnus Holmgren. If you have any comments etc, feel free to send me a note. You can reach me via internet on this address:

cmh@aug.s.se

Fido-net messages should go to "Magnus Holmgren",  
2:204/404.6@fidonet.org. Snail mail should reach me if you write the following address on the envelope:

Magnus Holmgren  
Rydsvägen 254 A:14  
S-582 51 Linköping  
SWEDEN

## 1.18 PicBoot.guide/Version history

Version history  
\*\*\*\*\*

Version 1.00

Version 1.01

Version 1.02

Version 1.03

Version 2.0

## 1.19 PicBoot.guide/Version 1.00

Version 1.00  
=====

Release date: 08 Feb 93

---

- \* Initial release.

## 1.20 PicBoot.guide/Version 1.01

Version 1.01

=====

Release date: 14 Feb 93

- \* PicBoot would sometimes read past the end of the picture-list file, although the file was correct. Also, the last picture would never be showed. Fixed (thanks to Daniel Joseph Oak for reporting).
- \* Did some tweeking, to make the program smaller. The total gain was eaten up by improved error messages though. PicBoot will now more accurately report why the display didn't open.
- \* Added support for masked ILBMs, since I discovered that I had such a picture anyway. :) Since I only have one picture, its hard to tell if it really works, but it seems to be ok at least.
- \* The autoscrolling of the screen didn't work. Fixed (so now you can view pictures larger than the screen, by moving the invisible mouse-pointer).
- \* Added the  
    CENTER  
    switch.

## 1.21 PicBoot.guide/Version 1.02

Version 1.02

=====

Release date: 21 Feb 93

- \* Added the DEFAULT switch. As Bradley Yen pointed out for me, there is (normally) only one monitor available early in the startup, and that is default.monitor (and what default.monitor is can vary. It can e.g. be PAL or NTSC). This switch will make the picture use the default.monitor, regardless what it was saved in.
  - \* Autoscrolling of screen disabled again, since this would make some overscanned pictures clipped, even if this wasn't needed.
  - \* A bug in the init code fixed.
-

## 1.22 PicBoot.guide/Version 1.03

Version 1.03

=====

Release date: 15 Oct 93

- \* If an error occurred, any black "cover screen" (BLACK option) wasn't closed. Fixed.
  - \* "Ported" to Macro68. At the same time, the code was cleaned up a little. All this saved more than 100 bytes from the program. Most of it eaten up by the improvements though.. :)
  - \* Added the MODEID parameter. rtGetModeID is included to make the use of this parameter easier.
  - \* Docs rewritten using Texinfo.
  - \* Pictures with mskHasMask were reported to have an "error in compressed BODY chunk", due to a bug. Sorry, but I don't have many pictures to test this code with.. :)
  - \* AUTOSCROLL switch added, so that you can scroll around in large pictures. Mostly useful when PicBoot is used from the Shell. During boot, this can cause the picture to be clipped when it isn't necessary.
  - \* PicBoot will now move the screen to the back before closing it (this makes the closing faster). Not needed during boot, but... :)
  - \* List files much improved! Each line will be parsed almost like the normal arguments. Old files should work fine, as long as no filename contains spaces, in which case those lines need to be quoted.
  - \* Altered command line syntax slightly. Needed because of the new listfile format. Tech note: I'm using a poorly documented feature of AmigaDOS: The /T specifier. If the argument following the keyword is YES or ON, the switch is turned on, if it is OFF or NO, the switch is off. Any other arguments causes an error. Testing revealed the true nature of this specifier... :)
  - \* Removed the "You need OS 2.04+" error message. An (Exec) Alert will be shown instead.
  - \* PicBoot will now random-select among the FILES arguments, even if the LIST argument have been used (previously, only the first file was used).
  - \* VIDEOOVERSCAN switch added. Makes the visible area as large as the system can handle. Requested by Stefan Boberg.
  - \* Screen centering improved. Should work fine with any screen mode now (but if you promote the screen, PicBoot will probably get it wrong. Unless the promotor also "promotes" QueryOverScan() :).
-

## 1.23 PicBoot.guide/Version 2.0

Version 2.0

=====

Release date: 29 Mar 94

- \* BLACK argument removed. Not needed any more, since PicBoot now will first allocate the needed memory, decode the picture into this memory, and then open the screen. This makes the screen opening/closing a little faster too (practically instaneous on my A4000/040).
  - \* Pictures (or rather, brushes) that were less than 16 pixels wide wouldn't decompress properly... :)
  - \* Added support for the CMAPOK flag in the BitMapHeader.bmh\_Flags (previously called bmh\_Pad) field (if this flag is set it indicates that the color map contains 8 bits/color rather than 4 bits/color).
  - \* Major code cleanup. Made the program somewhat larger, but... :)
  - \* GIF support added. Should be a little faster than PPSHOW.. :)
  - \* The  
MODEID  
argument wasn't properly "passed on" to any following listfile(s).
  - \* Rewrote rtGetModeID into GetModeID, that first checks for Asl, and then tries with ReqTools before giving up. This new version is in C, compiled with DICE, without any startup code, and is fully residentable. :)
  - \* Included UnpackILBM, that takes any IFF ILBM file (with a BODY chunk, i.e. a normal picture) and writes it with an uncompressed BODY instead. Written upon user request..
-