

Mostra

A Universal IFF Viewer
Version 1.09

by Sebastiano Vigna

1 Introduction

Mostra is an IFF ILBM viewer that can process *any* IFF ILBM file; this means not only IFF ILBM FORMS, but also nested ILBM, FTXT, FORMS, CATS, LISTS and PROPS. It has become reknown in the Amiga community for both its robustness and its flexibility.

Any design requires tradeoffs: **Mostra** tries to give you a good balance of program size (less than 15K), speed (optimized assembler decompression routines) and features. It tries to be enough smart to have always reasonable defaults, but also lets you force any aspect of the display, such as resolution and video flags. While it is not possible to foresee every possible race condition, **Mostra** should be able to cope with any reasonable ILBM file.

Mostra currently relies on ARP v39 in order to work. Thus, the `arp.library` file has to be in your `libs:` directory if you want to run **Mostra**. Otherwise, a message will be displayed, and **Mostra** will refuse to run. (Without ARP's powerful functions, **Mostra** would have never been released.) However, **Mostra** 2.0 will work only under Release 2 and will not need ARP.

One of the major features in this release is support for multi-palette pictures, i.e., pictures with the new PCHG chunk which should substitute the old CTBL and SHAM chunks. See the `Changes.doc` file for additional informations. SHAM is still supported, but the support of CTBL pictures via external call has been dropped since it was shown to be unreliable.

The other major improvement is complete support for the AGA chip set.

2 Syntax

Mostra can be called both from the CLI and from the Workbench. It is also a pure program, i.e., it can be made resident.

2.1 Command Line Usage

To get the command line syntax in AmigaDOS style, type:

```
M ?
```

This will produce the input template:

```
Files/...,A=All/S,R=Repeat/S,Q=Quiet/S,NoFastDraw/S,
C=Center/S,B=BlackBackground/S,W=Width/K,H=Height/K,Cycle/S,
Secs/K,Fade/K,NO=NoStartup/S,Batch/K,LockPic/S,
N=NoMouse/S,F=FreeMouse/S,LockKeys/S,NA=NoActivate/S,DB=DoubleBuffering/S,
Hires/S,Lace/S,Lores/S,NoLace/S,HAM/S,Halfbrite/S
```

If you ask for more help by typing '?' again, you will see:

```
Mostra 1.09 © 1990,1991,1992 Sebastiano Vigna
Usage: M <wildcards [...] | !> [All] [Repeat] [Quiet] [NoFastDraw]
[Center] [BlackBackground] [Width n] [Height n] [Cycle]
[Secs time] [Fade speed] [NoStartup] [Batch file] [LockPic]
[NoMouse] [FreeMouse] [LockKeys] [NoActivate] [DoubleBuffering]
[Hires | Lores] [Lace | NoLace] [HAM | Halfbrite]
```

The same help is displayed if you call **Mostra** with no arguments.

Mostra's first argument is a list of patterns (or file names) as long as you want; you will get every picture whose name matches one of the patterns.

Alternatively, you can type only the '!' character, and **Mostra** will display the ARP file requester, allowing you to search for the picture you want to show.

2.2 Workbench Usage

When **Mostra** is called from Workbench without any argument icon, it will pop the file requester and will allow you to select pictures to show until you select **Cancel** or close the file requester window. Alternatively, you can **SHIFT**-select multiple icons together with **Mostra**'s one, or you can change the default tool of your pictures. For details about this, see Section 3.2 [Startup Tool Types], page 5. Note that if you select an IFF CAT or LIST you will have an automatic little slideshow.

2.3 Keyboard Usage

Once you are looking at the picture you selected, you can exit by moving the mouse pointer to the very top of the screen and pressing the right button (you can also press the **RETURN**, **SPACE** or **ESCAPE** keys). You can drag or depth arrange the screen with the left mouse button (the gadget(s) are invisible, but they really are there!). **TAB** will toggle on/off color cycling. To stop pattern matching or IFF scanning use **CTRL-C** at any moment.

If the screen size is less than the size of the picture, you can move around with cursor keys alone (moving by 8 pixels) or in combination with the **SHIFT** (a screenful minus 16 pixels), **ALT** (which brings you to the borders), or **CTRL** (16 pixels) keys. A noteworthy feature of **Mostra** is that the picture is *not* decrunched all at one time in a very large chunk of memory; instead, a fast LM routine decrunches the file in real-time every time you move (the routine could be faster, but it has lots of controls to get rid of damaged IFF BODY chunks; safety always first). This is not as fast as the memory-eating method, but I think it's *much* better. Very large pictures (like 800x900) will be shown in a full screen (generated on the fly at Workbench size) which you can move around. Overscan pictures will retain their size if it's not greater than 362x283 lo-res pixels (362x241 on a NTSC Amiga). For hi-res pictures, there is also a 704 pixel width limit due to Intuition (724 under Release 2). Small brushes will be shown in a minimum 256x128 pixel screen. Of course, moving vertically with multi-palette pictures involves a high system overhead, since Copper lists are continuously built and deleted.

A weird effect of this method is that you can see pictures that can't even be *loaded* into your Amiga. Let's suppose you have a really crunched picture; the BODY chunk and a 256x256 lo-res screen could take much less space than the decrunched picture, so you could create pictures that only **Mostra** could let you see in all of their parts. Why you would want to do this, well, that's another matter.

2.4 Options

This list describes all the options available in **Mostra**. The letters between parentheses denote availability of a shortening. Options can be put anywhere on the command line, or in Tool Types (see Section 3.2 [Startup Tool Types], page 5).

A11 (A) forces a recursive scan of all of the subdirectories during the wildcard search.

Repeat (R)

will endlessly restart from the first pattern when it has finished with the last; if you used the ! line arguments, the file requester will pop up after every picture until you **Cancel** or close.

Quiet (Q) turns off all messages except in case of error.

NoFastDraw

leaves the program at priority 0; by default, **Mostra** will surround the **BODY** decrunching operations with a **SetTaskPri(1)**. This will not lock the system, but in a multitasking situation **Mostra** will perform quickly when needed.

Center (C)

will force the screen to be centered (640x200 pictures on PAL screens look so ugly. . .) and will use overscan if necessary. Note that if you move the screen, and you are not under Release 2, the centering will go away.

BlackBackground (B)

will create a 0-bitplane black screen that will stay until you saw the last picture. This is for aesthetic purposes only (note that no Chip memory will be used).

Width (W)**Height (H)**

want an argument that specifies a dimension of the viewing screen. However, **Mostra** is an intelligent program, and will almost always find the correct screen by itself. **Warning:** dimensions allowed are from 64 to 1024, at your own risk. Your Amiga won't crash, but weird things can happen. . .

Cycle will automatically activate color cycling on every picture. Usually this happens only by request, through **TAB**.

Secs must be followed by a number between 0 and 1000 (the zero delay was a request from a BIXen). **Mostra** will display each picture for the number of seconds specified, but you can skip using the mouse or the keyboard as usual (see Section 2.3 [Keyboard Usage], page 2) or stop the show with **CTRL-C**. Cursor keys are disabled. Greatly reworked on Warren's request.

Fade lets you specify a speed for pictures to fade in and out. Valid numbers are 1 to 4, with four the slowest. Note that HAM pictures can't be faded.

NoMouse (N)

will hide the mouse pointer while **Mostra** is displaying a picture.

FreeMouse (F)

will let you pass from one picture to another by clicking either of the mouse buttons at any position on the screen.

LockKeys will discard any input from the keyboard during the display. You must use the mouse to pass to the following picture. Note that if **FreeMouse** is not activated, the only way to go is clicking the right mouse button while the mouse pointer is at the top of the screen.

LockPic will lock the picture displayed. You won't be able to scroll around, but the memory in which the file is loaded will be immediately freed after the decompression, thus minimizing the memory usage.

NoActivate (NA)

will inhibit the activation of the picture screens, so you can keep on typing on your current screen. This was a request from Tom.

DoubleBuffering (DB)

sets a double buffering show mode: while a new picture is being loaded you will see the old one, no Workbench pop-up or black screens. It eats lots of memory, too. Again, a request from Tom. (Note: it's canceled if you call the file requester.)

Hires

Lores

Lace

NoLace

HAM

HalfBrite

force all screens to be in the respective mode (note that if **Mostra** finds a six bitplane picture with no **CAMG** chunk, it won't set the **HAM** flag by default). These flags are dumb, that is, they will do exactly what you say, even if it is meaningless. However, if you use both **HalfBrite** and **HAM** together, only the former will take place.

The **NoStartup** and **Batch** options will be discussed later, in Section 3.1 [Startup Files], page 5.

2.5 Examples

And now, some examples:

```
M MyPics:* Secs 5 Fade 1 NoMouse BlackBackground Repeat Center All
```

will generate a slide show of the pictures in the **MyPics:** directory and in all its subdirectories, centered and with no mouse pointer hanging around. In order to stop the slide show, you must press **CTRL-C**.

```
M df0:*.image df0:pictures/*.pic Hires Lace
```

will show the files ending with **.image** in the directory **df0:** and the files ending with **.pic** in the directory **df0:pictures**. **Mostra** will be forced to use high resolution, interlaced screens. Setting the **Hires** flag on pictures with more than 4 bitplanes usually leads you to see absolutely nothing.

```
M dh0:hirespic Lores NoLace
```

will show a hi-res picture in lo-res (*zooming in*). You can move around with the cursor keys as described above.

Width and **Height** can be useful when you have a picture that cannot be displayed because there's not enough Chip RAM (you get the 'Can't open Screen' error message). In such a case, try:

```
M pic Width 128 Height 128
```

This will usually allow you to at least get a peek at something.

3 Advanced Features

For maximum flexibility, **Mostra** allows you to use *startup files* and *startup Tool Types*, with which you can configure the program to suit your tastes.

3.1 Startup Files

There is a standard startup file, called **S:Startup-Mostra**. **Mostra** will search for it when run from the CLI. The format of this file is *exactly* the same as the **Mostra**'s line format minus the command name. Commands may be spread out over several lines or gathered together onto a single line. Every switch or keyword in the startup file will act as a default, and will be toggled or superseded by any command line arguments. Command line switches will act as toggles (if your startup file has the keyword **All** and you say **M * All**, you *won't* go into subdirectories), while keywords simply assert the new value (if you have **Fade 1** in your startup file and you say **Fade 3** in the command line, your pictures will be faded at speed 3).

Two options concern startup files:

NoStartup (NO)

inhibits the search for startup files, useful if you have one and you want to specify your options from scratch.

Batch wants a complete path/filename that **Mostra** will use as startup file.

3.2 Startup Tool Types

From the Workbench side, you can write the startup options in the Tool Types of **Mostra**'s icon, and it will use them. The format is the same of **Startup-Mostra**. Please note that **Fade=1**, **Width=352** are valid, but **FreeMouse=ON** is not; use only **FreeMouse**.

Or you can set the first Tool Type of a project icon to **STARTUP** and put your options in the following lines; the project picture file will be shown with those options when you double-click its icon. You can do the same thing with an icon that has no related file, and put some wildcards in the Tool Type. The wildcards will then be shown with those options. If you don't put in wildcards, you get a *style* icon; you can **SHIFT**-click some icons, **SHIFT**-click the "style" icon (in *this* order) and then **SHIFT**-double-click **Mostra**'s icon (or directly **SHIFT**-double-click the last icon, if its default tool is **Mostra**): you will see the selected picture files with the options specified in the style icon. The style icon by itself will pop up the file requester. Please note that even an icon with wildcards can be used as a style icon because multiple selections supersede wildcards.

I know, it seems a little twisted, but using it you'll find, instead, that it's the way you'd think.

Warning: strange interactions can take place. If you have some picture icons, and you **SHIFT**-click all but one of them and then you **SHIFT**-double-click the last one *and* the last picture icon you clicked had some **STARTUP** options, you will see *all* of the pictures with those options. Right?

For curious/tech/interested people, here is the exact algorithm:

1. If `sm_NumArgs>1` (the user started `Mostra` with at least an icon argument), check if `sm_ArgList[1]` (the first real argument) has Tool Types with first Tool Type `STARTUP`. If so, scan the whole Tool Types list like a command line, accepting mixed commands on a single Tool Type as well as separated commands in separated Tool Types.
2. If the first step goes wrong, check the `Mostra` icon Tool Types, and take them as a command line. (Note: `STARTUP` is not requested.)
3. Now, if `sm_NumArgs>1`, check every `sm_ArgList[i]` with `i>0` and if there is a related file (*not* a `.info` file!) generate a list of arguments as if the user entered them manually, *superseding* eventual filenames/wildcards found in the previous steps.

3.3 Startup Examples

Now, examples, by means of common problems:

- “I have a hundred pictures on my 360M hard disk, and I’ve organized them in a few groups. How can I make simple slideshows, with each group separate?”

Simple: create a project icon for each group, and add a Tool Type `STARTUP`. Then add a Tool Type like `PICS:Group1/* All Secs 5 Center` to choose your options, set the default tool to `C:M` (or wherever you put it) and double-click (of course, this must be done with every group icon). Don’t give the icons the same name as a directory, or `Mostra` will collapse in confusion.

- “I want to always see the black screen when I start from `WB`.”

Set a `Mostra` Tool Type to `BlackBackground`.

- “I have some pictures. Sometimes I want to see a few of them with a full screen and no mouse, sometimes with a 128x128 screen, sometimes centered.”

Create three “style” icons. Each icon must have as its first Tool Type `STARTUP`, and the following ones must be something like `NoMouse`, `Width 128 Height 128` and `Center`. The default tool must be `C:M`. When you want to see in a certain style, `SHIFT`-click the picture icons and *then* `SHIFT`-double-click the style icon you desire.

- “I like to click my pics one at a time, each one with different options. I’d like also to see them in irregular groups, each picture with its options.”

You need a psycho-analyst, not `Mostra`.

4 Acknowledgments

I wish to thank some people, first of all the ARP team for the `arp.library` and the SAS team for the best Amiga C compiler. Speed and compactness of `Mostra` come from the fabulous 5.10b release. Also I would like to thank the people that encouraged me somehow or helped me to find bugs, in particular Steve Tibbett, Warren Block, Tom Rokicki, Christopher A. Wichura and Loren Wilton.

If you’re reading this, you probably have access to a `TEX` system. Anyway, if you spent a good part of your time preparing and printing documents, I’d suggest you to take a look at `AmigaTEX`, a wonderful package from Radical Eye Software which is probably the best implementation of `TEX` you can find on any computer.

Mostra is © 1990,1991,1992 Sebastiano Vigna and it's not public domain: it is shareware. You should send me \$20 (or more, why not?) if you find it useful. Please don't send checks drawn on a US bank, they're almost impossible to cash. I'd prefer postal money orders, currency or checks drawn on an italian bank (in this order). **Mostra** is freely distributable as long as all of its files are included in their original form without additions, deletions, or modifications of any kind, and only a nominal fee is charged for its distribution. This software is provided "AS IS" without warranty of any kind, either expressed or implied. By using **Mostra**, you agree to accept the entire risk as to the quality and performance of the program.

Comments, complaints, desiderata are welcome.

5 Author Info

Sebastiano Vigna
Via California 22
I-20144 Milano MI

BIX: svigna
INTERNET: vigna@imiucca.csi.unimi.it
 vigna@ghost.sm.dsi.unimi.it
UUCP:cbmehq!cbmita!sebamiga!seba@cbmvax.cbm.commodore.com
 ...{uunet|pyramid|rutgers}!cbmvax!cbmehq!cbmita!sebamiga!seba
FIDO: 2:332/607.28

Table of Contents

1	Introduction	1
2	Syntax	1
2.1	Command Line Usage	1
2.2	Workbench Usage	2
2.3	Keyboard Usage	2
2.4	Options	2
2.5	Examples	4
3	Advanced Features	5
3.1	Startup Files	5
3.2	Startup Tool Types	5
3.3	Startup Examples	6
4	Acknowledgments	6
5	Author Info	7