Mostra

A Universal IFF Viewer

Copyright © 1990,1991,1992,1993 Sebastiano Vigna

Version 2.0

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 is well renowned 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, 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.

Version 2.0 has been completely rewritten with the Release 2 in mind (and with an eye towards 3.0). This means complete AGA support, localization, clipboard reading, display database access, selectable autoscroll, no need for arp.library, and more. The support for SHAM images has been dropped, because the new PCHG chunk supports much better line-by-line palette changes.

2 Installation

`Mostra` does not need any particular treatment when installed. However, if you are interested in localization, you should take some care.

Localization means that `Mostra` will try to give its messages in your preferred language. It is a feature available under 2.1 and beyond. For it to work, `Mostra` has to be able to find out its *catalogs*. There are two ways of doing this: either you copy the '`Catalogs`' directory coming with `Mostra`'s distribution in the same directory where `Mostra` lives, or you copy the contents of the '`Catalogs`' directory in your '`Locale:Catalogs`' directory. The first solution allows for easier maintenance, but with the second one you can move `Mostra`'s executable anywhere, and you will never have any problems. If you decompress `Mostra`'s archive, the '`Catalogs`' directory will be automatically created for you.

3 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.

3.1 Command Line Usage

To get the command line syntax in AmigaDOS style, type 'M ?' followed by RETURN; this will produce the input template:

```
Mostra 2.0 Copyright (C) 1990,1991,1992,1993 Sebastiano Vigna
Files/M,Clip/K/N,A=All/S/T,R=Repeat/S/T,Q=Quiet/S/T,
NoFastDraw/S/T,C=Center/S/T,B=BlackBackground/S/T,
Cycle/S/T,Secs/K/N,Fade/K/N,NO=NoStartup/S/T,Batch/K,
LockPic/S/T,N=NoMouse/S/T,F=FreeMouse/S/T,LockKeys/S/T,
NA=NoActivate/S/T,DB=DoubleBuffering/S/T,DM=DisplayMode/K,
DC=DisplayClip/K,CAMG/K/N,W=Width/K/N,H=Height/K/N,
AS=AutoScroll/S/T,HiRes/S/T,LoRes/S/T,SuperHiRes/S/T,
Lace/S/T,NoLace/S/T,HAM/S/T,HalfBrite/S/T,FallBack/S,
SMR=ScreenModeReq/S,FR=FileReq/S,PubScreen/K
```

(If you do not know how to read a template, take a look at the AmigaDOS manuals.)

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. The second argument specifies instead the number of a clipboard unit to read (this will usually be 0). Note that you *can* specify both arguments on the same line.

Alternatively, you can type only the 'M' character, and Mostra will display the system file requester, allowing you to search for the picture you want to show. Note that under version 2.1 and beyond you can select multiple files using SHIFT-click.

For a detalied explanation of the keywords, see Section 3.4 [Options], page 4.

While showing each picture, Mostra displays some information about the file and the screen mode. First all the information IFF chunks (such as ANNO, AUTH, *et cetera*) are printed, then the picture and the display clip dimensions, followed by the type, depth and size of the screen, are displayed.

3.2 Workbench Usage

When Mostra is called from Workbench without any argument icon, it will pop up the file requester and will allow you to select the pictures to show. 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 4.2 [Startup Tool Types], page 8. Note that if you select an IFF CAT or LIST you will have an automatic little slideshow.

When Mostra is called from Workbench, it never outputs diagnostic or error messages. If you have problem in displaying a picture, you may want to try from the CLI. This can raise a problem if

the 'Repeat' option is used, and 'BlackBackground' is not specified (see Section 3.4 [Options], page 4). In this case, if none of the specified files and icons is an IFF picture, since no window will be ever opened Mostra will continue to endlessly scan the argument list, and the only way of stopping it will be by sending a break signal using the status and the break commands from the CLI, or some task handling utility.

All the options available on the command line are available through tool types. See Section 3.4 [Options], page 4, and Section 4.2 [Startup Tool Types], page 8. You can also peek at the tool types of Mostra's icon.

3.3 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. SHIFT-TAB will do the same, but it will also reset the palette to its initial value. 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 assembler 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) in which you can move around. Overscan pictures will retain their size of it's not greater than the video overscan available for the picture video mode (for instance, 362x283 lo-res pixels in PAL, or 362x241 in NTSC). Small brushes will be shown in a minimum 128x128 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.

If you own a large amount of Chip memory, you can also consider using the 'AutoScroll' keyword, that will force Mostra to decrunch the whole picture in one gulp, and to use the system screen scrolling features.

3.4 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 4.2 [Startup Tool Types], page 8).

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

`Repeat (R)`
: will endlessly restart from the first picture when it has finished with the last one; this includes repeating the wildcard search, checking for the clipboard, and opening the file requester, if any of these options has been specified. If you used the `FileReq` line argument, the `Cancel` gadget will break the loop.

`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. If your program has already a priority greater than one, `Mostra` will leave it untouched.

`Center (C)`
: will force the screen to be centered. Note that the centering is based on your Overscan preferences.

`BlackBackground (B)`
: will create a 1-bitplane black screen that will stay until you saw the last picture. This is for aesthetic purposes only (note that almost no Chip memory will be used).

`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. `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 3.3 [Keyboard Usage], page 3) or stop the show with `CTRL-C`. Cursor keys are disabled.

`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.

`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. (Note: it's canceled if you call the file requester.)

`DisplayMode (DM)`
sets a specific display mode from the graphic database. For instance, you can ask to display your picture in '`NTCS:Lores`'. Note that under 2.1 and beyond the database entries are *localized*, so that you need to know the name of the display mode in the first language currently selected in the locale preferences. This option is the safest and most precise way of forcing a display mode (but see also the `ScreeModeReq` option).

`DisplayClip (DC)`
sets a specific display clip from the graphic database. For instance, you can ask to display the maximum available amount with '`VIDEO`'. Allowed values are '`TEXT`', '`GRAPHICS`', '`MAX`' and '`VIDEO`'. Note these entries are *not* localized. The '`VIDEO`' option, in particular, allows you to see 736 pixel width pictures in PAL and NTSC (see also the `ScreeModeReq` option).

`CAMG` sets a specific display ID, expressed as a decimal number. This option is mainly useful for developers. It has the same effect as a CAMG ILBM chunk.

`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. The minimum allowed dimension is 128; smaller values will be silently increased to this value. Note that if '`AutoScroll`' was specified, only the display clip dimensions are changed—the screen will retain its full size.

`AutoScroll (AS)`
sets the automatic scrolling feature. The whole picture has to be decompressed in a big chunk of Chip memory, but then you can scan through it very smoothly by moving the mouse towards the edges of the video (exactly like in a Workbench which is bigger than your screen size). It works also with multipalette pictures (but in rare occasions, some glitches may be visible in certain scroll positions).

`HiRes`

`LoRes`

`SuperHiRes`

`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. You should avoid in general the use of these options, and rather ask for a specific display mode with '`DisplayMode`', '`DisplayClip`' and '`ScreenModeReq`'.

`FallBack`

forces `Mostra` to try to guess the best display mode with which it can display a picture in case its natural (or user specified) display mode is not available.

`ScreenModeReq (SMR)`

forces `Mostra` to open the screen mode requester on the public screen specified by the '`PubScreen`' option, allowing you to specify a display mode, display clip, screen size and optional autoscroll. This option is functional only under 2.1 and beyond. Note that the selections made in the requester will supersede all the related keywords (such as '`Width`', '`AutoScroll`' *et cetera*). However, if the keywords are specified they will be used in order to initialize the requester fields.

`FileReq (FR)`

tells `Mostra` to open the system file requester on the public screen specified by the '`PubScreen`' option. Note that you can specify this option together with file names and clip numbers. `Mostra` will turn its attention to file names, clips and input from the file requester, in this order.

`PubScreen`

specifies a public screen to open the file and screen mode requesters on. The screen is moved to front before opening the requester. This option is functional only under 2.1 and beyond.

The `NoStartup` and `Batch` options will be discussed later, in Section 4.1 [Startup Files], page 7.

3.5 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 DisplayMode PAL:Hires
```

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 screens.

```
M dh0:hirespic PAL:Lores AutoScroll
```

will show a hi-res picture in lo-res (*zooming in*). You can move around with the mouse.

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.

4 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.

4.1 Startup Files

There is a standard startup file, called 'S:Startup-Mostra'. Mostra will search for it whenever it is launched. The format of this file is *exactly* the same as the Mostra's command 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.

4.2 Startup Tool Types

From the Workbench side, you can write the startup options in the Tool Types of `Mostra`'s icon, and `Mostra` will use them. You have to fill each Tool Type with an option (previous releases would allow even multiple options on the same line, but this is in contrast with Commodore's guidelines). Please note that '`Fade=1`', '`Width=352`' are valid, but '`FreeMouse=ON`' is not; use only '`FreeMouse`'.

These options supersede or toggle the options specified in the startup file exactly as command line options. See Section 4.1 [Startup Files], page 7. If you have no startup file, this does no matter.

You can also add Tool Types to a picture icon, and they will supersede or toggle again the settings specified by the startup file and by `Mostra`'s icon Tool Types.

You can also prepare an icon with no related file, but having a Tool Type such as '`Files=PICS:#?`'. Double clicking on the icon will show all the pictures in '`PICS:`'.

The previous version of `Mostra` had a feature called *style icons*, by which a selected pattern of preferences, contained in the Tool Types of an icon, was applicable to a series of icons. This is no longer possible (and the need for the '`STARTUP`' keyword is gone), but under Release 2 you can easily create a hard link to `Mostra`, and give it a new icon with new Tool Types, and a '`Files`' Tool Type. This will produce the same effect.

4.3 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 like '`FILES PICS:Group1/*`', plus all the tool types you would like to apply to the pictures; then 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 a series of hard links to `Mostra` (they behave like copies of the program, but they take almost no space on your disk). You can give icons to the hard links, and put different options in the Tool Types of each icon. Then, using different icons will start `Mostra` with different settings.
- "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`.

5 Acknowledgments

I would like to thank the people that encouraged me someway or helped me to find bugs, in particular Warren Block, Tom Rokicki, Steve Tibbett, Christopher A. Wichura, Loren Wilton and all the other beta testers, which are really too many to mention.

The catalog files were provided by Reinhard Spisser, Stefan Becker, and Martin Taillefer.

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 © 1990,1991,1992,1993 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 currency, postal money orders 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.

6 Author Info

```
Sebastiano Vigna
Via California 22
I-20144 Milano MI
```

BIX: svigna@bix.com
INTERNET: vigna@ghost.dsi.unimi.it
UUCP: seba@sebamiga.adsp.sub.org

Table of Contents

1 Introduction 1

2 Installation 1

3 Syntax 1

3.1 Command Line Usage 2
3.2 Workbench Usage 2
3.3 Keyboard Usage 3
3.4 Options 4
3.5 Examples 6

4 Advanced Features 7

4.1 Startup Files 7
4.2 Startup Tool Types 8
4.3 Startup Examples 8

5 Acknowledgments 9

6 Author Info 9