

MacroSystem

V L A B M O T I O N

Motion JPEG
Video Board with
MovieShop Software

for the
Commodore Amiga

Users Manual

Contents

| | | |
|----------|---------------------------------------------------------------|-----------|
| 1 | What is VLab Motion? | 5 |
| 2 | Installation | 7 |
| 2.1 | What You Need | 7 |
| 2.2 | Installing The Hardware | 8 |
| 2.3 | Installing The Software | 8 |
| 3 | The MovieShop Software | 10 |
| 3.1 | Getting Started | 10 |
| 3.2 | The Principles Of MovieShop | 11 |
| 4 | The Windows and Menues of MovieShops User Interface | 16 |
| 4.1 | The Windows | 16 |
| 4.1.1 | The Scene Window | 16 |
| 4.1.2 | The Control Window | 18 |
| 4.1.3 | The Record/Play Window | 19 |
| 4.1.4 | The Define Source Window | 21 |
| 4.1.5 | The Video Options Window | 22 |
| 4.1.6 | The Advanced Options Window | 23 |
| 4.1.7 | Color Keying - The Color Settings Window | 24 |
| 4.1.8 | The Block Window | 26 |
| 4.1.9 | The Clipboard Window | 27 |
| 4.1.10 | The Movie List Window | 27 |
| 4.1.11 | The Macro Window | 30 |
| 4.1.12 | The Audio Window - Simultaneous Audio Using the Toccata Board | 31 |
| 4.2 | The Menues | 33 |

| | | |
|----------|------------------------------------------------------------------------------------|-----------|
| 4.2.1 | The Project Menu | 33 |
| 4.2.2 | Edit | 34 |
| 4.2.3 | Windows | 34 |
| 4.2.4 | Stamps | 35 |
| 4.2.5 | Settings | 35 |
| 5 | Using the VLab Motion As An Animation Recorder | 38 |
| 5.1 | The multipic.library | 40 |
| 5.2 | The JPEG Direct Import | 41 |
| 6 | Exporting The Video Images And Digital Archiving | 43 |
| 6.1 | Exporting Video As A Scene | 43 |
| 6.2 | Exporting Video As Separate Images | 44 |
| 7 | Macros and ARexx | 47 |
| 7.1 | General Information | 47 |
| 7.2 | A Short Description of Different Parameters | 48 |
| 7.3 | Return Values and Error Codes | 48 |
| 7.4 | The ARexx-Commands Of <i>MovieShop</i> | 50 |
| 7.5 | ARexx Application Example: Using <i>VLab-Motion</i> With <i>Scala MM</i> | 86 |
| 8 | Future Plans For <i>MovieShop</i> | 89 |
| 8.1 | Digital Effects | 89 |
| 8.2 | Audio Editing | 90 |
| 8.3 | <i>MovieShop Pro</i> And Hardware Modules | 90 |
| 9 | Troubleshooting | 92 |
| A | Literature List | 94 |
| B | Report | 96 |
| C | Hotkeys | 98 |
| C.1 | Menue-Shortcuts | 98 |
| C.2 | Controlling <i>MovieShop</i> With The Keyboard | 98 |

Chapter 1

What is VLab Motion?

VLab Motion is what people call a killer video product. It supplies features many users have been waiting for since the dawn of time.

Being a Zorro-II-add-in card for the Amiga 2000, 3000 and 4000 computers, VLab Motion digitizes video directly to hard disk and is able to play the video back from hard disk. It turns your Amiga into a tapeless video recorder! This is even more amazing if you keep in mind that the video is top-quality full motion, meaning 50/60 frames per second in full YUV 4:2:2 quality.

But a killer product needs more than that: VLab Motion can overlay video from an external source with hard disk-recorded still or moving video using its color-comparable embedded genlock. This opens the most common studio effects technique for desktop video users: Blue Boxing is the keyword. By defining a certain color range as being transparent, you can let the hard disk-recorded video shine through the now-transparent parts of the video coming from an external source.

For example: Let's say you want to let somebody walk through the desert. First, record a desert landscape to your hard disk. Then let the actor walk in front of a white (or blue, green, etc.) wall. Declare the wall color as transparent. Wow: The video on the VLab Motion outputs show the actor walking in the desert.

Now, imagine what kind of effects you can do using this method! Hollywood has used it (and is still using it) for all kinds of special F/X. Lets see if you can outwit them!

But back to the basics: VLab Motion is recording the video by digitizing it and further by compressing the enormous amount of data (27 MB/sec.) into JPEG files on the fly, bringing it to recordable data rates of 200 KB-2.5 MB/sec. The video is now stored compressed, but digital. The Amiga can access the video frame accurate and instantly. Therefore, editing the video is as easy as editing text - maybe even easier since you do not have to type it in.

VLab Motion's main software, the MovieShop program, is basically an editing system

for digital video. It directly controls the hardware, so all the digitizing and the playback is done from MovieShop. Using a hierarchical editing method, MovieShop enables you to digitize and reorganize your video with breathtaking speed and comfort. It is also able to import and export images to the Amiga for playing rendered animations or for adding digital effects (like Morphs) to the video you have recorded.

But what is video without audio? Most of the comparable solutions for digital video just support video and force the user to handle the audio using traditional tape-based methods. But since we are no longer in the 20s, where video was accompanied with piano music, we have made audio an integral part of the system as well. Using the Toccata audio board, a product that digitizes CD-quality audio directly to hard disk, in combination with VLab Motion, you can record audio and video simultaneously just like a video recorder does. No time code is necessary since simultaneous recording keeps the video and audio always lip-synchronous. If you are not used to other systems, this may not sound like a sensation, but it is actually a revolution at this price.

We hope that you will like our product. Keep in mind these are its early days, and both the manual as well as the software will be enhanced strongly in the very near future. We are nevertheless very proud of our product and we have little doubt that you will find it as amazing as we do. Please give us your input regarding the software, and the features you'd like to see included.

This manual will explain the system entirely. We are anyway assuming that you have some knowledge of the Amiga itself. We will certainly concentrate on giving you explanations and tips that will further your understanding of the product. Read at least the first three manual chapters completely - we promise we won't bore you too much. The software is pretty much self-explanatory once you are used to the idea of its operation!

So, thank you for purchasing VLab Motion. Have fun with digital full motion video - and please, get back to us if you want to report your experiences or detected bugs, or also if you feel you need to give us some applause. Keep in mind that this manual was translated directly from German, so if the jokes are too hilarious for you, now you know why.

As one of the early VLab Motion purchasers, you'll be receiving the first two software updates free of charge. Also, newer versions of this manual will be made available to you. Thanks for your patience.

Chapter 2

Installation

2.1 What You Need

For using VLab Motion, you need

1. An Amiga 2000, 3000 or 4000 computer with at least one free Zorro-II slot and at least 2 MB Fast RAM
2. The VLab Motion itself plus the MovieShop software that comes with it
3. A video source (NTSC or PAL) with CVBS or Y/C output
4. A video display monitor with CVBS or Y/C input
5. A free hard disk partition or, even better, an entire free hard disk

For getting the best possible results, it would be nice to have

1. A fast hard disk system, for best data rates either on the motherboard or on the CPU accelerator board (like the CSA Magnum 040)
2. A fast SCSI hard disk (cannot be fast enough or large enough)
3. 8 MB RAM or more
4. A Retina display board for instant display of images in full color and for running the software on high resolution, flicker free screens
5. A MacroSystem Toccata audio board for adding the audio side to VLab Motion
6. At least an 030 processor, or even better an 040
7. A heater that warms your naked feet after VLab Motion has knocked your socks off!!!

2.2 Installing The Hardware

To install the hardware, switch off the Amiga. Open the case and select a free Zorro II slot. Remove the bracket. Install the card by pushing it carefully into the slot until it fits. Tighten the screw at the bracket. Close the computer again.

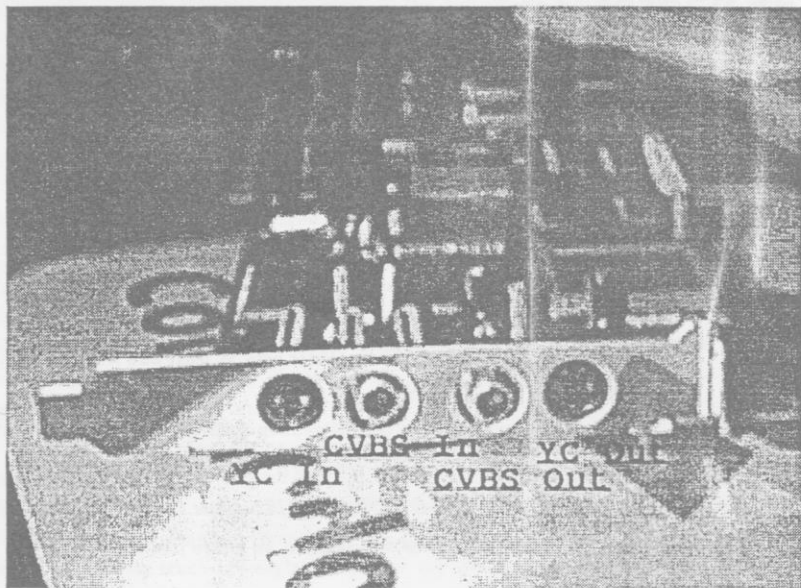


Figure 2.1: The Video - Connectors

Now connect the video cables. The VLab Motion has CVBS and Y/C inputs (one of each), and similar outputs. The inputs are on the slot side of the card (Y/C in, CVBS in, CVBS out, Y/C in, beginning from the slot side).

Close the computer. Pull your socks on tight. That's it!

2.3 Installing The Software

Install the software by using the supplied "Installer" script. It will copy all necessary files to your harddrive.

Remember, you're going to receive the first two upgrades free of charge.

The software consists of

1. The MovieShop software (anywhere you want)
2. The pfile.library (has to be in libs:)
3. The vmotion.library (has to be in libs:)
4. The multipic.library (has to be in libs:)
5. The VJPEG file (has to be in c: or in the program directory)
6. The movieshop.catalog (has to be in locale:catalogs/english, deutsch... whatever language you prefer)
7. The movieshop.prefs (has to be in envarc:movieshop)
8. The movieshop.macros (has to be in envarc:movieshop)
9. A text file containing 200 of the latest dirty jokes (may not be available in countries still regarding sex as dangerous and immoral)

1-3 are absolutely necessary while 4-8 (and especially 9) are just nice to have. If your system reports the lack of certain files, repeat the installation procedure or copy the above listed files to your system hard disk and that's it!

Chapter 3

The MovieShop Software

3.1 Getting Started

The MovieShop software is a unique video tool, specifically designed for VLab Motion. It controls the entire hardware device and even the hardware of the Toccata audio board. MovieShop is a non-linear video editor as well as a playback program for rendered animations, and furthermore it is a control program for the fabulous color compare feature for blue-box special F/X.

To start it, just double-click on its icon or use the CLI.

If one or more files are missing, you will get a report. If the prefs are reported as being unavailable, just go ahead. You will save your personal settings pretty soon.

The MovieShop GUI consists of many different windows, each one is responsible for a specific job. After the GUI is opened, you will not see all the windows immediately - if you do not have a high resolution display board such as the Retina, you may not even have enough space for all of them.

It is not necessary to keep all the windows open. The idea is to give all the options, but keep everything configurable by the user to reduce confusion. You can keep windows closed if you do not need them now or iconify them for instant access. You can set your personal customized interface up and change it whenever you want!

The first thing you will have to do is to set the video norm you plan to use. If you are a US-American or Canadian user, you will probably prefer NTSC over PAL. Choose "Define Source" under the menu "Windows" and pick the videonorm of your choice in that window. Then, open the window "Video Options" by clicking the gadget in the Define Source window. Set the output mode either to auto or to the video norm you want. Close the "Video Options" window again. Now, select the video input you plan to use (either Y/C or CVBS).

Now, the video on your TV monitor should already display the video you are feeding

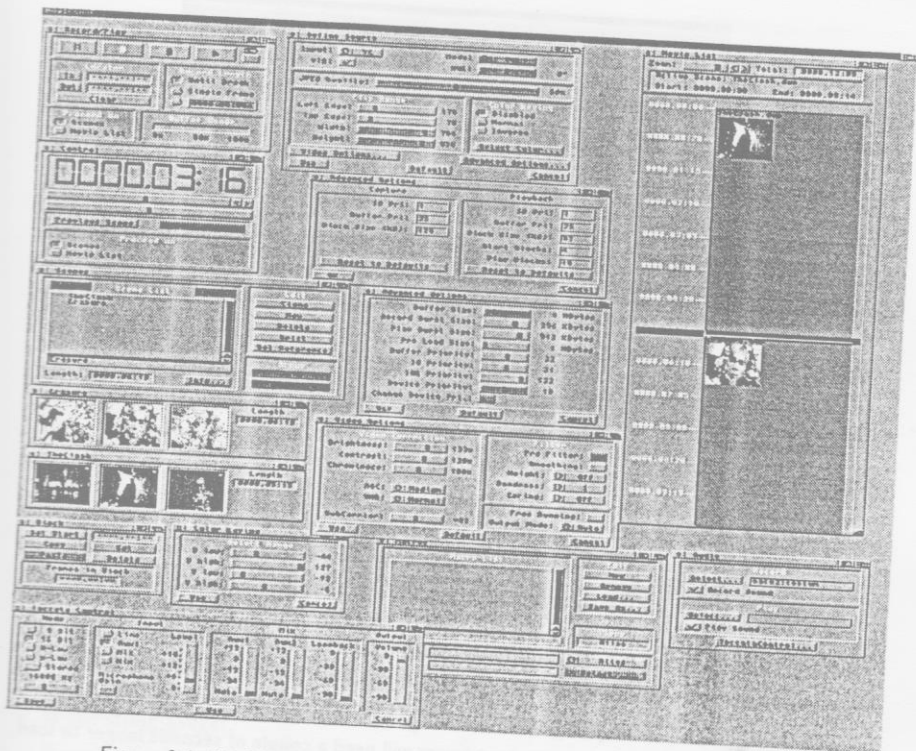


Figure 3.1: 80 % of the windows are open - a 1280 x 1024 screen

into VLab Motion. If that does not happen, check your cabling and the video settings. The video you see is digital video already! You have online control over hue (NTSC), contrast, brightness, gamma and more ... far beyond the control you can have on analog video.

It is already time for saving the settings you have fixed so far. Use the menu point Settings-SaveSettings-To Archive to do so. The next time you start the software, your settings will be similar again.

3.2 The Principles Of MovieShop

Although you are probably very keen to go on, you should take the time to read further on. It is necessary to learn something about the idea behind MovieShop to use it

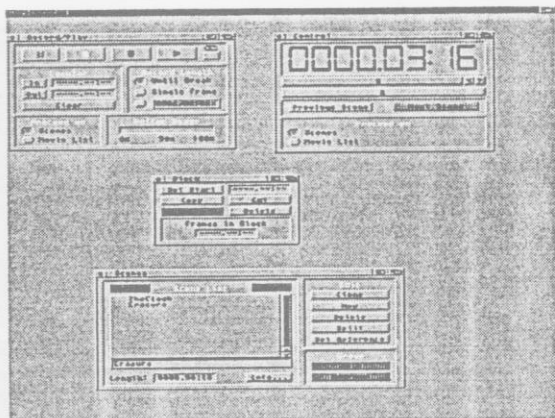


Figure 3.2: All you need to record, edit and play - 800x600 screen

efficiently.

First of all, you will have to dedicate a complete hard disk partition to MovieShop. Since the AmigaOS filesystem is not suited for this kind of job, MovieShop will control this partition by itself. This partition will not be readable by the Amiga except over MovieShop. All the contents of a partition you allow MovieShop to use will be lost. Make sure you are prepared!

Try to dedicate a fairly large partition on your fastest hard disk. It will probably not make much difference if your text processor will need a couple of seconds longer to load, but for MovieShop, a fast hard disk is essential.

MovieShop will store all the images directly into this partition. You can export and import images from outside, of course.

MovieShop is really a two-monitor system, displaying the user interface on the RGB Amiga (or Retina) screen, and displaying the video on a TV monitor connected to one of the VLab Motion outputs.

The normal way to edit a piece of video is

1. Digitize just roughly the parts of the video you regard as useful or import the video from Amiga image files
2. Fine tune these scenes by removing unwanted parts at the beginning and at the end, and if necessary within the scenes.
3. Reorganize the now-clean scenes.
4. If you want, add digital video effects to the video by using tools like image

processors.

5. Play the video back in its new organization and record it to your new master tape. If you want, save the video to an AmigaOS medium for further use.

These few steps may sound simple and plain. But of course, this is a very general explanation that covers very complex procedures. Be patient with your learning curve. You should expect this procedure to be time consuming and possibly a little bit frustrating. Remember you're dealing with amazingly powerful software working on a more than complex job.

The first thing you will have to do is to start a new project, very much like in a text processor. Select "Projects- New" from the menu.

Now, you will have to create your first scene. A scene can be just one image, but also an entire movie. The images that make a scene can either be digitized directly from VLab Motions video inputs or be imported from Amiga image files.

If you compare MovieShop with a text processor, you may say that images (frames) are MovieShops words, while the scenes are its sentences.

These scenes are listed and controlled in the "Scenes" window. You can generate a scene by recording video or by selecting a new scene in the window (for importing Amiga image files).

Digitizing video is done in a very simple, VCR-like style. Each video frame can be accessed directly. Each record automatically generates a new scene that appears in the scene window. It can be renamed by just clicking on it and then by changing the name in the string gadget.

If you are not happy with the positioning of the window, you can change it on the fly. Display a still image by using the slider in the 'Control' window and change the offsets in the 'Define Source' window. If you are unhappy with the resolution, start a new project and fix the width and height before digitizing.

You can now edit the scene by using the "Block" window. Just cut, copy, and paste the frames - again, similar to a text processor.

If you have fine tuned the scenes, you can now start to reorganize them. The reorganization is done in a hierarchical way, like generating a DOS file structure (tree). On your hard disk, you probably have a structure like "Work:Textprocessors/ProWrite" or even deeper. The same way to bring order into your video is used by VLab Motion! You can have a tree like "Holiday94 - Summer - Beach-Volleyball scenes". This makes it real easy to find your way through your scenes. The scene window will allow you to click through a multi-levelled video, like in a file requester.

To structure the scenes this way, you will have to use another window: The clipboard. If you want to "group" (like putting them in a common folder) two or more scenes, open

the clipboard window, click on a scene to be grouped (in the scene window) until the mouse pointer turns into a hand. Then, drag the scene into the clipboard and release the mouse pointer. Ready! The scene now appears in the clipboard. Do this with all the scenes you want to group. When the selection is complete, click on "Make Group". Drag the new group back into the scene window! Rename the new group now. That's it! If you want, you can create a group containing other groups and therefore generate hierarchical levels. The clipboard technique also allows you to change a scene or a group inside the hierarchical tree.

The groups are very important for MovieShops operation idea. If we go back to the text processor analogy, you may call the groups the chapters of a book - the chapter consists of sentences, the sentences are formed by the words of the text. MovieShops groups are formed by the scenes, and the scenes are formed by the single images (frames).

You may now begin to have a sense about the power of this method of video editing. If you want to add digital effects to certain parts of the video, you can now export a block of frames as Amiga files for further processing with any of the image processing programs. After you have performed the effect calculation, you can re-import the changed frames into MovieShop.

If you feel that you have finished the whole procedure, you may want to record the results back to tape. Keep in mind that the data on the MovieShop partition will be lost if you start a new project. You can of course re-digitize the edited video tape - but with a significant quality loss. Any time you leave the digital domain, quality losses are unavoidable! If you want to keep the data digital, store them on an Amiga medium such as a hard disk or a tape streamer tape using the export function (as a sequence of images) or save the main scene by using the 'Save Scene' function which saves the entire current scene or group as one huge file, only readable by VLab Motion.

At this stage, you can also use the blue-box feature for overlaying your edited digital video with external analog video and record the result back to a video tape. This is great for desktop special effects!

IMPORTANT: If you want to break your work and later pick up again, you can save the project and re-open it at any time. But keep in mind you can not re-open your previously saved project if you have worked on a new project after saving the previous one. The data on the partition will no longer be the same - opening a new project makes any previously saved project invalid. This is important!!

You should now have learned about the way MovieShop works. The next step in this manual will be a rough explanation about each window of the MovieShop user interface.

If you can't wait, go ahead and play with MovieShop. If you need explanations, you can refer back to the manual.

Chapter 4

The Windows and Menues of MovieShops User Interface

MovieShop's user interface has three major elements:

1. The windows
2. The menus
3. The ARexx interface

The first two listed elements will now be explained, while the third element is the focus of the chapter 7 on page 47.

It makes sense to run the MovieShop software parallel to the reading of this chapter, continously crosschecking if you understand the window or menu with all its options.

4.1 The Windows

4.1.1 The Scene Window

The scene window lists all the video you have imported, either by digitizing it from video or by importing it from Amiga image files. It really works as the platform for all further editing, as the hierarchy navigator of the MovieShop system.

In the list view gadget, the scene window lists all the scenes available in the selected level (remember the hierarchy idea ?).

Plain scenes are listed with its name, groups are listed with an arrow in front of them. You can go into a group by double-clicking on its name. You can go into any eventually existing higher level by clicking on the "Parent" gadget.

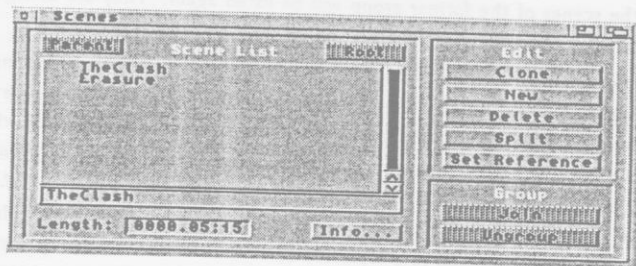


Figure 4.1: The Scenes - Window

You can select one of the listed scenes or groups by single-clicking on it. It will be displayed in the text gadget below. You can change the name of the selected item by using the keyboard in the normal way - return confirms. The length of the scene will be displayed in minutes, seconds and frames. Clicking on another scene will bring that one into display. The scene you have selected here is called the current scene. This selection is important, since all the edit events and also the play events are performed on the current scene.

The gadgets in the right part of this window give you control over the listed scenes. The changes you do to the video are performed on a pointer list. Therefore, they are very fast. This method is called non-destructive editing. You can instantly clone a scene, for example, and delete this one if the edits that you have done to it are not what you wanted. The original scene will not be affected. A click on "Clone" creates a new scene, named the same as the original with a ".dup" as appendix. A click on "Delete" will delete the current scene again.

The gadgets "Split" and "Set Reference" will need another window to make sense - the "Control" window. Using this one, you can move through the current scene with a slider. Releasing the slider displays a still image. If you click on "Set Reference" now, this still image will represent the current scene from now on. If you select this scene later again, the reference frame will immediately appear on the video monitor.

If you click on "Split", the scene will be divided into two halves, using the displayed frame as the break point. Two new scenes are created, replacing the original scene. The new scenes will adopt the name of the original one, adding a "- 1" and "- 2" as appendix.

The two gadgets titled "Group" are ghosted if the current scene is not a group. You can join the scenes of a group by clicking the "Join" gadget. The arrow disappears and the scenes of the group lose their independence. One branch of the hierarchy tree is cut off!

"Ungroup" will also delete the tree structure, but instead of creating a new independent

scene, the scenes of the former group will be listed again. They are brought one step higher in the hierarchy structure.

The last gadget to be explained is the "Info" gadget. This opens a new window, where the reference frames of the current scene is displayed. Selectable in the "Settings" menu, you will either see thumbnail equivalents of the reference frame or the first, the last and the reference frame. Since the image has to be calculated (dithered, scaled, decompressed), this can take a while. It gives you instant recognition of the scenes. You can keep the info window open if you want, any change on the displayed frames will be noticed and will lead to a replacement of the thumbnails. Many users will never use the info windows since they eat space on the user interface screen, and the video monitor displays the reference frame anyway. But other users will want to have the image visible on the user interface as well.



Figure 4.2: Two Info - Windows

4.1.2 The Control Window

This window was already mentioned before. It moves through the current scene or through the movielist (will be explained later). You can just use the slider or the jog-shuttle control (may be deactivated in early versions), but the cursor keys will work, too. The position in the video will be represented by the position of the cursor and by the time information in the top line.

You can set the size of the time information display in three steps (small, medium, large) using the settings-menu.

You can select other scenes in the same level by clicking on "Previous Scene" or "Next Scene".

If you have used the slider or the cursor keys, you will see a still image on the TV monitor. The quality might look poor compared to the running video, for two reasons: The display is only showing a field (halfframe), so the resolution is only 50 %, and

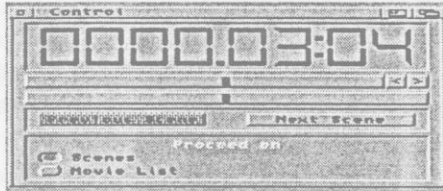


Figure 4.3: The Control - Window

the human eye is far more sensitive for noise and blurring when looking at still images. Anybody who has used a frame grabber knows about this!

If you want to see the video on the external input again, you will have to click the Stop gadget (square) in the "Record/Play" window.

4.1.3 The Record/Play Window

This window is pretty obvious. It controls the VLab Motion video digitizer/compression engine for bringing in video frames and further plays video controlling the VLab Motion decompression and display engine.

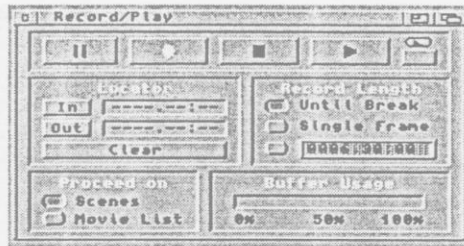


Figure 4.4: The Record - Window

This window should remind you of the interface on a VCR - it is indeed the digital equivalent of such a device. From the left to the right, you will find

- Pause (two vertical lines) Clicking on this gadget activates it. If you now press Play or Record, all the buffers are prepared for immediate start. If you release the Pause by clicking it again, the Play or Record is performed immediately. This is necessary for precise performance.
- Record (white filled circle) This button starts the recording. It generates a new

scene that will be the current scene automatically. The amount of recorded data is set in the "Record length" gadgets.

- Stop (black filled rectangular) This gadget stops a Record, a Play or a Display (control-window) operation and will show the running external video on the TV monitor again.
- Play (black filled triangle) The current scene or the movielist is played until the scene ends or the user clicks on Stop. Activating the loop gadget to the right will start from the beginning if the end is reached (may not work in early versions).

You can also set marks here if you just want to play a certain part of the active scene or movielist. The locator settings allow frame-precise playback of marked parts, starting with the "in" time and ending with the "out" time.

You can select whether the functions in this window will affect the current scene or the movielist, either by using the gadgets or by hitting the space bar.

The next part to be explained is the buffer utilization bar. It gives you visual control about the Play and Record performance of your system. To explain this, we will now do an excursion into MovieShops buffer technique.

First, you should know that JPEG is a compression system that does not give a stable data rate. You can set the algorithm to a certain factor, but the size of the compressed image depends strongly on the image itself. Detailed pictures with hard contrasts will give several times larger JPEG images than low-contrast images with few details, such as a setting sun. So, the data rate of JPEG video can jump up and down with scene changes. Since the hard disk data rate is also unstable, especially in a multitasking computer, the Record or Playback would often break when the data rates exceed the hard disk performance at that time.

To bridge over bottlenecks in the hard disk performance or over peaks in the video data rate, MovieShop uses a RAM buffer. The played data will not be copied directly to the VLab Motion, but first into a RAM buffer. This buffer will be filled if the performance is higher than necessary at that time and therefore serves as a reserve when the situation changes. Lets say you are recording a heaven and sea scene at a data rate of 1 MB/sec. Further, lets imagine your hard disk is good enough for 1.2 MB/sec. The buffer can be built up with 200 kB/sec now! If now a new scene gets digitized with many details and hard contrasts, the data rate jumps to, lets say, 1.5 MB. The buffer technique will not break immediately, since the buffer is full with reserve data and probably brings you over the critical scene. The more RAM you dedicate to this buffer, the smoother the video will be.

The buffer also works for the Recording, but here, the system tries to keep it as empty as possible. If the hard disk can not store the incoming data in time, the unrecordable

data will be put into the buffer. If the data rate now decreases, the buffer will be drawn back as fast as possible.

The buffer adds great smoothness to the system, especially when you work with slower, non-DMA hard disks. If you have enough RAM, you can record and play video with performances that you have never expected.

To get an idea about the performance of the system, the buffer status is displayed in the window. Remember: The play buffer should always be as full as possible, the Record as empty as possible. So do not be confused!

Refer also to the buffer settings explained in the Define Source/Advanced Options chapter.

4.1.4 The Define Source Window

This window gives you control over specific functions of both software and hardware. You can first of all select the video standard and the active input. If you have a running video source connected to this input, VLab Motion will genlock on this source, using the frequency from it and leading it through on the VLab Motion outputs. If no video is connected, VLab Motion will generate its own frequency (free running output). This video signal will probably be better than the one your VCR gives.

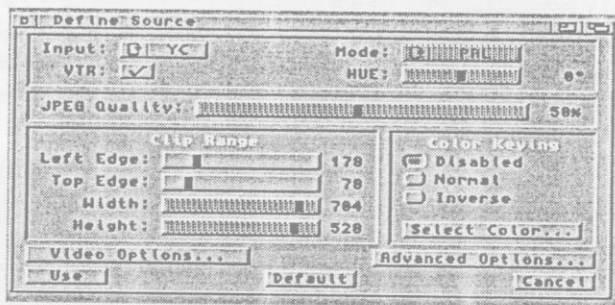


Figure 4.5: The Define Source - Window

Regarding VCRs: VLab Motion does not require a time base corrector unit. If you genlock (digitize) from a VCR, just activate the "VTR" switch and the video signal will be stabilized. This may not sound spectacular if you are not experienced with desktop video, but it saves at least the equivalent of \$ 800 for an otherwise useless TBC.

One of the most important parts of this window is the setting of the JPEG factor. It goes from 1 % to 100 %.

The higher you set the JPEG factor, the better the video quality will be, but also the higher the required disk speed and space will be. It is not possible to give an average MB/sec. data rate since this depends so much on the video you use. You will have to find your personal optimum, which you will have to change when you use different video. Set it high enough to achieve the desired quality, but low enough to give smooth Record and Play performance. Watch the buffer utilization display in the Record/Play window for fine tuning.

In this window, you can further set the resolution and the offsets of the video. The hardware supports full resolution, square pixel images (768*592 in PAL, 640*480 in NTSC). It is not likely that any video source you feed in does use this resolution. A TV monitor would not even display such video entirely. So, you can set offsets for both vertical and horizontal direction and the vertical and horizontal resolution. You can therefore select a rectangular window in the video signal that you want to use and save a lot of disk space by leaving non-visible parts of the video aside. The constraints of the JPEG algorithm will not allow free definitions though, so some of the settings will be rastered in necessary steps.

4.1.5 The Video Options Window

When you have connected a running video signal to the selected input, the video appears on the TV monitor and looks like it is being simply lead through. No big deal, right?

Well, the thing is, the video is not just being lead through. It gets digitized and re-converted for output on the fly at a data rate of 27 MB/sec. Therefore, you have direct control over the video. Or, to be more concrete: You watch the video the same way as the compression engine "watches" it.

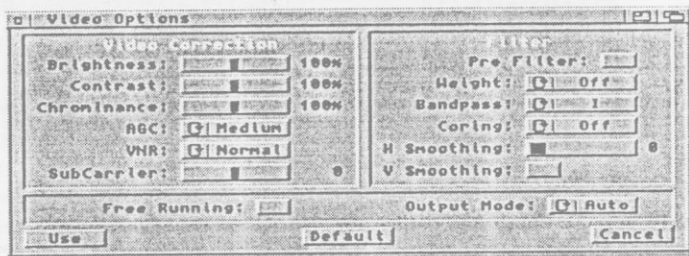


Figure 4.6: The Video Options - Window

This gives you great control over the video signal. You can change things like brightness, contrast, chrominance and use several filters to make the video look like you want it to look. The results affect the displayed video immediately! If it is too dark - try to raise

the brightness. If you miss color - raise the chrominance. Also, try the different filters offered. If the video flickers in color, you may change the SubCarrier setting. It really depends on the video you are feeding in how the filters should be set, but generally, a CVBS signal needs more filtering than a Y/C signal.

You can also choose the desired video norm here. If you set it to "Auto", it will always use the video norm of the signal fed in or stored in your scenes.

Another important item is the smoothing, now available for horizontal and vertical directions. Due to the nature of the JPEG algorithm (refer to the suggested literature in appendix A on page 94 if you want to know more), the JPEG compression is more effective if the image is noise-free and soft-edged. Try to work out the highest smoothing that gives you enough details - you can possibly raise the JPEG quality without requiring more datarate!

4.1.6 The Advanced Options Window *RIGHT AMIGA 'E'* *"OPENWINDOW ADVOPTS"*

The Advanced Options give control over several settings related to the buffer and also over priorities related to playback and I/O tasks.

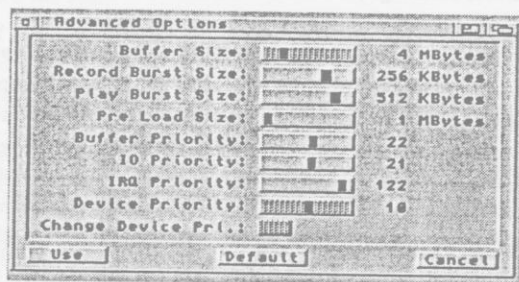


Figure 4.7: The Advanced Options - Window

Most of these settings have practical defaults, figured out by many of our testers. If everything works fine, leave it as is. If you are having problems while recording or playing back, we recommend that you refer to the "troubleshooting" chapter for detailed tips.

The main value you may want to change is the buffer size. As mentioned before, a large buffer adds smoothness to the performance of recordings and playbacks. If you have enough RAM, you should raise it to a pretty large size - the larger, the better. But leave some RAM free - you may need it for other applications running in background.

The burst sizes (separated for read and write) set the amount of data to be stored or read to and from hard disk. These sizes may have various ideals that are related to both

the hardware and driver software of your hard disk controller. The more multitasking-friendly the hard disk system performs, the larger the ideal value will be. If you have problems in the form of buffer overflow or underflow, try to change these settings.

The pre-read buffer sets the amount of data that MovieShop pre-loads into the buffer before the playback starts. As already mentioned, the buffer bridges over data rate peaks in the video and over hard disk bottlenecks, e.g. in case the hard disk recalibrations. A large pre-read will give a smoother display when such a situation happens very soon after the playback starts, but will delay the playback for the time needed to fill the buffer.

The priorities are settings controlling the multitasking-related behaviour of involved tasks. They are very critical. Wrong settings can make the whole procedure go mad. We recommend that you leave the priorities as they are and only play with them if nothing else works in case you suffer from problems.

4.1.7 Color Keying - The Color Settings Window

The color keying feature is one of the most amazing parts of the VLab Motion. It can be used to create stunning special effects right on your desktop!

The color keying procedure overlays 2 video signals. One is digital, one is analog. The basic idea is that one video signal is defined as being partially transparent and the other signal is shining through, so that the result is a video signal comprising a mixture of both original signals. To make real blue-boxing possible, the definition of transparency is the key word. Each pixel of the video needs to be defined being either transparent or non-transparent! If you realize that the video can have 450,000 pixels, you might understand why there was no solution for affordable desktop systems until now.

But as you know, VLab Motion digitizes the video in real time and outputs it again. All the video settings affect the video immediately. The basis for the color keying feature is also performed by the digitizer engine of the card: The color parts of the video are analyzed on the fly.

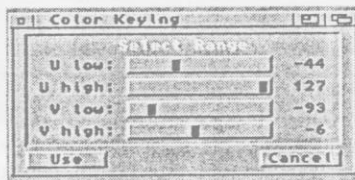


Figure 4.8: The Keying - Window

The color values for the transparency can be selected by the user. But if you are familiar

| Color | U low | U high | V low | V high |
|---------|-------|--------|-------|--------|
| Red | -44 | -11 | 84 | 105 |
| Green | -71 | -38 | -90 | -66 |
| Blue | 75 | 112 | -33 | 0 |
| Yellow | -104 | -60 | 0 | 27 |
| Cyan | 5 | 46 | -95 | -74 |
| Magenta | 43 | 75 | 70 | 92 |
| B/W | -57 | 40 | -36 | 32 |

Table 4.1: Value Suggestions For The Colorkeying

with the data sheets of the VLab Motion (and/or with digital video in general), you will remember that the video is not in RGB form but in YUV (4:2:2). YUV is a storage method far more efficient than RGB since it matches the video technique much better. The video idea is that the human eye has a very good monochrome (black and white) resolution and a weak color resolution. Video now works like a child's paint book: The outer borders of objects are very sharp, but black and white, and the color fills the objects. The result is a colorful image that looks very sharp - the low color resolution does not disturb the eye.

The Y in YUV represents the monochrome part of the video - b/w video contains only Y parts. U and V are the color part of the video. Only the latter have an influence on the color keying.

The definition for the color keying is a little complex - it needs to be so. You can not just say "make red transparent", since even a solid red background will have shades and lighter parts. So, you will have to define ranges instead.

The other thing is that you are working with U and V data. It is not possible to guess the values of U and V that you need for a special color like you could expect when working with RGB. You need to figure out the color range using trial and error. This is pretty easy, though.

Lets pick up the example of the running man in front of the red wall that you want to overlay with a desert landscape that you have taped on a VCR.

First, digitize the landscape to hard disk and edit the scene completely until it fits your needs. Display one of the images using the "Control" window. Now, play the man-in-front-of-red-wall video (or use your camera online). Set the Color Keying in the "Define Source" window to "Normal". Open the "Select Color" window. You may already see a kind of flickering - this is already the desert video shining through. Now, drag the sliders for "U high" and "V high" to the extreme right and the "V low" and "U low" into the extreme left position. You will now only see the displayed man in front of the wall. Now, carefully drag the sliders towards the middle until the wall (and only the

wall) gets transparent. Thats it! Clicking on "Invert" in the Define Source window will make the desert shine through the man - not useful here, but sometimes necessary and interesting.

If you now click on "Play", the video scene with the running man will have the desert instead of the red wall as background - the guy is really a jogging freak, right?

If the camera signal should disappear when you start playing: Check the video options. You may have the free running function active! Deactivate it to genlock on the camera signal - this is required for the color-keying.

As already said, we can not give scientific equivalents for the colors and the appropriate settings in the window. But here is a list that could give you some start points:

Please do not regard this table as being set in stone. Again: You will have to use trial and error for the color keying, but please don't forget how many hours people spend in Hollywood studios to set up a blue box based special effect.

4.1.8 The Block Window

The block window is responsible for the heart of computer-style editing. Here, you can cut, copy, paste single frames or whole sequences and therefore totally re-arrange the video.

All the edit events do not physically move the frames stored in the MovieShop hard disk partition. They just change the list MovieShop runs for each scene, so editing is blazingly fast using this non-destructive editing method.

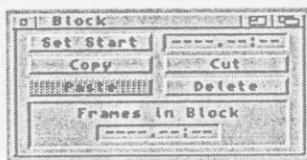


Figure 4.9: The Block - Window

To edit, just drive the scene to the desired first frame that you want to cut or copy and click on "Set Start". Now, you can use the Control window again to choose the last frame affected. Clicking on "Cut" removes the selected part into the buffer, "Copy" just copies it into the buffer, but leaves it in the scene as well.

You can now select a different scene or even create a new one, or just select a new position in the same scene. Then, use "Paste" to put the contents of the buffer into the new position. Ready!

The buffer contents can also be exported as Amiga image files using the menu "Edit-Export Block".

As already explained, the block starts with the first displayed frame and ends with the last displayed one. This is different to text editors that usually start with the line after the current one. But in a text processor, you see at least 20 lines on the screen - here, you just see this image. So, we have decided that it is more efficient to also cover the current frame.

If you paste, the buffer frames will be inserted BEFORE the displayed image. The displayed image will then follow the last inserted frame.

4.1.9 The Clipboard Window

The Clipboard window can group or ungroup video scenes and groups. It therefore creates new hierarchy levels further navigated through by the Scene window.



Figure 4.10: The Clipboard - Window

To group scenes, just click on a scene in the Scene window. Keep the left mouse button pressed. It will turn into a hand. Now drag the scene into the clipboard listview gadget and release the mouse button. You have moved it into the Clipboard! Do the same with any other scene you want to put into this group.

If you have finished, just click on "Group". The scenes are now grouped! Copy them back into the Scene window or move them into the Movie List using the same method as before.

The other purpose of the Clipboard window is to move scenes or groups into other hierarchy levels.

4.1.10 The Movie List Window

The Movie List is basically a window that contains all the scenes to be played back in the right order. Other systems call it the time line.

You do not need the time line for playing the video back using the order you want. You

could arrange everything in the Scene window and totally forget that there is the Movie List window.



Figure 4.11: The Movie List - Window

But there are several reasons why the MovieList is important. First, you have a better overview about the video in terms of playback order. Second, there is a pointer line going through the time line when you use the Control window to drive through the video. This gives you visual control over the position in the overall timing. Third, future versions of MovieShop software will give you the option to physically reorganize the hard disk partition, using the Movie List as the master for the order.

The third point is pretty important in case you have performed massive editing. The internal list will then contain many jump marks, so the hard disk will be forced to step a lot. This reduces the playback performance significantly - a hard disk that steps can't read.

One of the coming software updates will anyway bring video and audio together in the MovieList.

Scenes or groups can be copied into the Movie List using the same click-drag-drop technique described for the Clipboard-window. You can do this with scenes and groups from both Scene and Clipboard window. Scenes can be reorganized inside the Movie List using only the mouse, and of course at any time put back into the Clipboard or the Scene window.

Thumbnails (stamps) of the first or the reference frame will appear in the MovieList if the sizes allow. At the top left of the window, there is a slider for zooming. This means you can see all the Scenes inside the Movie List in the window, probably without stamps (there will be no room for them). The cursor line will move very slow since all your video is displayed in the Movie List window.

You can zoom into the List at any time.

All the Play and Control related events can be directed to the Movie List by clicking the gadgets in the Record/Play and Control window, or by hitting the space bar (which toggles between Scene and Movie List mode).

If you drive through the video using the cursor keys or the sliders in the Control window, you will see how the cursor line reacts.

In the settings menu, you have several Movie List related options, all affecting the stamps. You can set if you do or do not you want stamps, or if you only want stamps that have already been generated before.

4.1.11 The Macro Window

If you plan to use your *VLab-Motion* a lot, the *MovieShop* software will soon be your daily workplace. It is like the word processor if you'd be a book author!

Experienced users often work out standard procedures that can get pretty complex. Many mouseclicks and/or menu operations may be required for such procedures.

Because of this known effect, we have invented the Macro window. Here, you can put any *MovieShop* command or even a full set of commands (in a script) on a key of your choice! You can even control other programs with Arexx compatibility directly using this powerful feature.

For example, you could take the current frame, cut it, save it to RAM:, load it into ADPro (running in background), sharpen it using ADPro's convolve filter, save it again and import it into *MovieShop*, right in the position where the original image was. All this could be done using one keystroke!

You can define macros in the window by editing an existing macro (just click into the

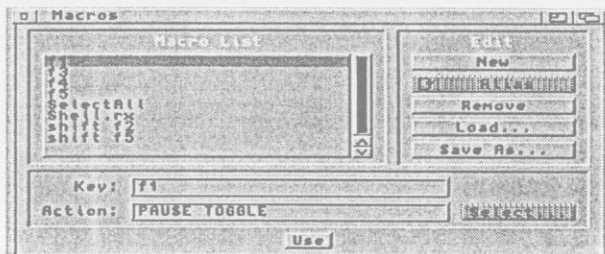


Figure 4.12: The Macro - Window

string gadgets), by defining a new macro or by loading a set of previously saved macros in.

If you define a new macro, you can decide if you want to use it as an alias (just one ARexx command, to be typed into the bottom string gadget) or if you want to use a script file loaded from external.

Complex procedures will require scripts, indeed. You will have to get used to ARexx to fully utilize the *MovieShop* macro feature! But this is anyway a good idea – without knowing ARexx, you certainly loose a lot of oportunities in general.

You can define any key you'd like, f.e. F-keys or combinations of keys. Your AmigaOS manual (refer to the 'F-Key' chapter there) will explain the available combinations.

Even ARexx-haters will be able to put simple things like Play, Record and Stop on keys! Just click 'New', choose 'Alias', click in the upper string gadget named 'Key' and type 'F1'. Then, click in the lower string gadget named 'Action' and type 'PLAY'. That's it! Click on 'Use' and try if pressing F1 will play the scene currently selected in the Scene window.

You have the choice of the full set of ARexx commands (refer to chapter 7 on page 47 for these)! It is certainly worth playing with it.

You can of course save and load your set of definitions. They will also appear in the 'User' menu as executable commands.

4.1.12 The Audio Window - Simultaneous Audio Using the Toccata Board

As you may know, MacroSystem has not only developed many video products, but also the Toccata 16 Bit Audio I/O board, a Zorro II card offering 16 Bit stereo audio with CD-sampling rates and hard disk recording.

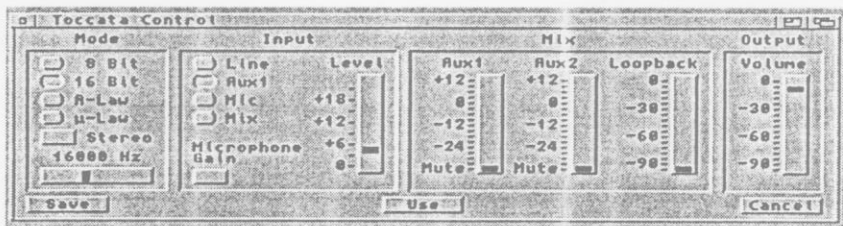


Figure 4.13: The Toccata Control - Window

Therefore, it should not be a surprise that MovieShop software fully supports both cards for digitizing and playback of video and audio at the same time.

Of course, you will not be able to use the audio-related features of the MovieShop software if the Toccata is not installed in your system.

The principle of audio recording in the MovieShop software is that with the recording of each scene, an audio file will be stored in an AmigaOS directory. This file will be named the same as the scene, with the appendix ".snd". It contains the precise lip-synchronous audio.

When playing back a scene, an audio file that carries the right name will be played with it.

It is possible to use the strong Samplitude software that comes with the Toccata board to edit the audio file.

If you rename the scene in MovieShop, the audio file will also be renamed automatically. Note that editing the video does not affect the audio, but any event that results in a loss of the scene (like splitting it) will lose the audio, not by deleting it, but by losing the name relation.

You can also give a name for an audio file to be played with the MovieList.

The settings of the Toccata - like inputs, frequencies, levels and more are done inside the ToccataControl program. This serves as a preferences program for the Toccata.

Although the Toccata software is completely independent, we have prepared MovieShop to allow seamless combination of both audio and video. You can open the ToccataControl program directly on the MovieShop screen, using MovieShops Audio window to open it. Since it has the same Style-Guide-compatible look, it fits 100 % into the GUI of MovieShop.

To start the ToccataControl window from the Audio Window of MovieShop, you will have to set the path where the ToccataControl is stored. You can do that using the "Settings- Pathes- ToccataControl Path" menu point. A file requester will allow you to

select the ToccataControl program for further use. Save the MovieShop settings after you have done this.

Also specify a directory for the audio files this way.

You can now activate the audio record and playback independently by checking Play and Record in the Audio window.

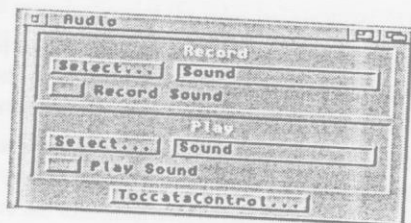


Figure 4.14: The Audio - Window

Save the MovieShop settings again.

At this point, we would like to assure that coming versions of *MovieShop* will be able to edit audio and video together, entirely in *MovieShop*. This is in preparation right now and will ease the procedure a lot.

4.2 The Menues

MovieShops Menues are responsible for things like settings, load/save procedures and setups. Some functions from windows are implemented as well, launchable by hotkeys – for example edit events like cut, copy and paste.

We will not discuss every menu point in great detail, for the boringness of such didactics. We will furthermore try to give overviews and conceptional explanations that should make you understand the menu. Let's go ahead...

4.2.1 The Project Menue

This menu is easy to explain. You can open or start a new project (remember the old project is lost if you use the same partition!), you can close a project, save your work, save scenes as one AmigaOS file (refer to chapter 6.1 on page 43), get information about the current project and the software version – and, last not least, quit the program.

Important is the Project Info. Here, you can see the length of the video and the occupied harddrive space, so you can calculate the average data rate of your project. By using a stable image (like a fixed camera), you can determine the maximum datarate of your

system. Just start a new project, record 10 seconds, divide the project size through 10 and you got the rate! Now start a new project and raise the JPEG quality until the record buffer overflows while recording (do not use too much buffer for testing).

You can also set the optimum block sizes using this method.

4.2.2 Edit

The Edit menu gives you control about the frames in your project. You can import images from an AmigaOS partition (refer to 5 on page 38, delete frames, perform all the block operations like cut, copy, paste, export a block and control the clipboard.

Most of the functions should be pretty obvious here, but we would like to mention the 'Paste Reverse' function extra. This can be used to achieve a ping-pong like effect! Just copy the sequence you would like to ping-pong into the buffer and paste it reverse. New frames will be written to the harddisk, because otherwise, the harddisk would be too slow – it is not prepared to step backwards.

Also remarkable is the Clipboard function, because it can do things quicker than the mouse-related window method. You can move all the scenes and groups from the Scene window into the Clipboard and the other way round. If you have to move like 10 scenes, you win a lot of time!

Note that you have to select the necessary options for importing frames and exporting the block in the Import and Export Settings windows.

4.2.3 Windows

This menu is the navigator for the user interface design.

As you know, the *MovieShop* interface consists of many independent windows. These windows can be opened from here. Each main window is listed and can be opened by selecting it. If it was already open, it will be put to front, in full size in case it was iconified.

There are still many windows that cannot be opened from this menu. These windows, like the Video Options, will have to be opened from other windows like the Define Source. Only the core parts have menu entries.

You will probably see the window 'YUV Settings' as ghosted. This window controls the YUV component output module, which contains many broadcast standard settings and controls. It can only be opened if *MovieShop* has detected its presence.

You can also execute a command here, this menu point opens a small text window where you can type ARexx commands in. This is great for testing ARexx commands

without writing a script and helps a lot if you use the powerful ARExx interface of *MovieShop*.

4.2.4 Stamps

Stamps are the small reproductions of the video images displayable in the Info windows and in the Movie List.

You can have an influence on the colordepth of these stamps by selecting a screenmode for the *MovieShop* screen (do that in the Settings-Menue).

The Stamp menu gives you general control about appearance and quality of the stamps and also allows you to take control over the memory needed for the stamps.

First, you can select if you want the stamps dithered and/or interpolated (raising the quality, but requiring more rendering time), Dithering can virtually raise the colorshades of an image. It is rendering new pixels into the picture to trick the human eye. This is very common in digital image processing and is probably not new to you.

The interpolation affects the way the stamps are downscaled from the original full-size fields. If interpolation is chosen, it uses sophisticated routines for the scaling, otherwise it uses the much quicker pixel-jump-method.

You can decide if you want to delete rendered stamps in memory if the info window is closed (Release on window close). If you select this method, re-opening the info window would give the rendering delay once again.

For the Movie List, you can select if you want stamps at all, and if you always want them or only if they already exist (in Info windows).

Finally, you can release all stamps to clear the aquired memory.

4.2.5 Settings

This is a very important menu in *MovieShop*, giving you finest control over elements of user interface and the operation modes of *MovieShop*.

It also needs more explanation than the other menus, beeing far more specific.

The settings will therefore be explained in more detail.

- Save Icons

This sets if you want to have icons saved with images, projects and scenes. This raises the necessary harddrive space, but gives you visual control if you use the Workbench rather than the CLI.

- WB Open

This sets if the Workbench screen should be closed (if possible) in order to free memory for *MovieShop* or if you want the Workbench to stay open.

- Select Displays

This affects the display operation mode. It sets if a still image is displayed as odd or even field. If you are suspicious that you are suffering from a bad JPEG image, you can change the display mode if you can not find it using the cursor keys (you are probably looking at the wrong fields).

You can further decide if you want to see stamps for the first, last and reference frame in an Info window or just for the reference frame.

- Position Marker

The position marker in the Movie List window can be small or large, just the way you like it better.

- Digit Size

Here, you can set the size of the digits in the Control window in three steps or go for the automatic mode. Again, it is up to you.

- Move Scene Delay

If you click on a scene in the Scene window, the Clipboard or the Movie List, the mouse pointer turns into a hand after a while. You can then move this scene. This setting sets the delay time between click and turn – just select your preference.

- Path / File Names

Here, you set the AmigaOS paths necessary for the *MovieShop* operation. The most important path is the partition for your video! Any partition larger than 2.5 MB will show up as available here, even a recoverable RAM disk (RAD:). Make sure you use a partition cleared for *MovieShop*! Also important is the path for the (Toccata) sound. You will have to specify a directory for the simultaneous audio. Choose a folder on a different harddrive than the video drive for best results – otherwise the stepping would slow the overall speed down massively.

Also you can set the path for the ToccataControl program. You can then open this program directly on the *MovieShop* screen.

- Screen Type

Here, you can set the screenmode used for the *MovieShop* screen. You can use the predefined screens, use the screenmode requester or pick one of the amazing Retina 16 or 24 Bit screens.

The main advantage of the high color screens (256, 65535 and 16.7 million) is the much better appearance of the stamps.

- Window and Screen Font

The *MovieShop* user interface strictly follows the Commodore User Interface Styleguide. Therefore, it is fully font sensitive. You can use any font your AmigaOS knows about. The windows and menus will automatically change the sizes to meet the new space requirements.

- Language

MovieShop also fully supports the locale.library, the amazing language-sensitive method AmigaOS invented with V 2.1 of the Workbench. If you have your AmigaOS running in German, the software will come up as German immediately. If it is English – you'll get English. Here, you can even take use of the alternative languages we implemented if you are owner of WB 2.0. The language is easily selectable. If you prefer a language not supported and want to do a translation, just contact us. If you send in a catalog file in your language, we would be happy to compile it for you and others.

- Safety Requests

Your video data is as valuable as any other work you possibly invest hours and days of work in. So, *MovieShop* is prepared to give warnings anytime you are in danger to destroy your previous work. But especially if your experience grows, you will soon get annoyed by all the requesters, literally paralyzing your creativeness.

Following the philosophy to give a maximum of flexibility, *MovieShop* gives you three levels of safety request intensity. If you select 'Many', you will get a lot of warning. 'Some' will warn you if it gets serious, and 'None' will not bother you with safety stuff at all. It's up to you!

- Load and Save Settings

All your settings like windows, positions, video settings, advanced options, paths, just anything can be saved for future reference. You can even save and load different settings (e.g. one for PAL and one for NTSC), using the 'Load/Save As..' function.

If you want to save the settings until the next change, no matter if you down the machine, save the settings to archive. They will be used automatically at every startup.

Auto-Save means the settings will be saved to archive whenever you quit the program. Quitting and Re-starting will leave the settings unchanged then. This is the recommended method!

Chapter 5

Using the VLab Motion As An Animation Recorder

VLab Motion plays back JPEG-compressed video images in real time. So, it has the potential for playing rendered images as long as they are in JPEG format! We have listened to many animators and followed their feedback reports. So, the current version of MovieShop is not only a perfect video editing software, but also a valuable and necessary tool for those users aiming at real-time 24 Bit animation playback.

The main difference between video and animation is that the video is digitized from external video sources, while animations are digitally created from the Amiga, now being stored as image files in many different resolutions and file formats.

As a difference to other available boards, we assumed the following basics:

- The animator wants to use the *VLab-Motion* not only for new projects, but wants to bring his previous work over as well.
- Animations will probably be saved as frames rather than fields. They might be rendered in a resolution not suitable for VLab Motion. They might have been stored in IFF 24 Bit ILBM or any other non-JPEG file format.
- The animator does not want to re-render the entire animation.
- The animator does not want to use an image processing software for preparation due to the complicated setup and long rendering time.
- The animator wants to load the files in one and not click each frame.
- The animator has the images probably stored in many different folders and used to edit the animation with FRED, the frame editor that comes with ADPro and MorphPlus.

- The animator may want to re-edit the animation and probably mix it with video scenes.

We did take care of these basics very much.

The concept of bringing animations into *MovieShop* is to set the import settings to the way you have saved them and then load these images into a *MovieShop* scene. Then, *MovieShop* will treat this scene as normal video, giving you exactly the same control as over true video scenes.

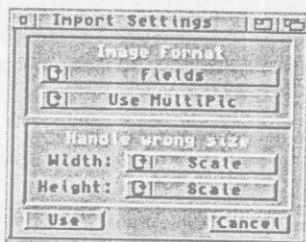


Figure 5.1: The Import Settings - Window

The nature of the images need to be set in the Windows-menus 'Import Settings'. Here, you can specify if you want to import frames or fields, if the data needs to be converted from AmigaOS image files (multipic) or if they are just the right format (JPEG quality ratio and resolution are already correct). You can also make *MovieShop* importing from a FRED sequence list (FRED-list, ending with '.seq'). You can also decide what *MovieShop* should do if the image is not stored in the resolution you use for the *VLab-Motion* (only if you use the multipic load). The options are

1. **Scale:** The image will be scaled to the image size for *MovieShop*.
2. **Center:** The image will be centered.
3. **Cut/Fill:** The image will be placed to the top left corner. Missing or undisplayable parts will be filled with black / cut off.
4. **Repeat:** The image will be repeated to fill the full resolution – nice for creating backdrops from brushes.

These different ways of adapting wrong resolutions can be set for both horizontal and vertical directions.

If you have done your settings, click 'New' in the Scene Window. A new, empty scene is born! Use the menu point 'Import Frames' to load the images. If you want to import a numbered sequence, click on the first image. *MovieShop* should detect the nature of

the image (sequence) and asks you about the last frame in a second requester. There you go! The images will be loaded, converted and stored automatically.

If you have set one of the FRED-import-modes, just choose the FRED-list in the file requester.

If your animation is loaded completely, you have it stored in the new scene! You can now edit it, group with other scenes, put it in the time line and of course play it back in real time.

If you want to play a prepared audio file with it using the Toccatà, just save the audio into your sound folder (refer to chapter 4.2.5 on page 35 to set the path for audio files) for *MovieShop* and name it the same as the scene, adding '.snd'. This audio will now be played with the animation! Remember activating the audio playback in the Audio Window.

Note that rendered animations can be very detailed. So, watch your data rate! If you can not play the animation from harddrive, consider lowering the JPEG quality in 'Define Source' and redo the import procedure.

Now, we have mentioned the multiple mode for the import settings. We will now explain what this mode does for you.

5.1 The multipic.library

We at MacroSystem are focussed on digital video and images. We carry no main product that does not have to load and save image files. So, we have invented a new concept of loading and saving images effectively: The multipic.library. This library easily allows to load and save multiple image formats, just using the comfortable library function calls. This saves harddrive and memory space and extremely speeds up development time.

The multipic.library is also used by other MacroSystem programs, like AnCoS and RetinaDisplay. It contains in a condensed form all necessary routines to load and save images in multiple file formats. Interested programmers can get (mostly free) licenses directly from MacroSystem.

Currently supported file formats are:

| Fileformat | type |
|------------|----------------------------------------------------------------------------------------------------------------|
| BMP | 1-8, 24 Bit (uncompressed only) |
| IFF DEEP | 24 Bit RGB, RGBA [TVPaint, VLab] |
| IFF ILBM | 1-8 Bit (2-256 colors incl. EHB) HAM (4096 colors) HAM8 (262144 colors) 12 Bit 24 Bit (incl. CLUT) |
| IFF RGB8 | 24 Bit [TurboSilver, Imagine] |
| IFF RGBN | 12 Bit [TurboSilver, Imagine] |
| IFF YUVN | YUV-411, YUV-211 and gray scales [VLab, ...] |
| PGM | 8 Bit gray scales |
| PPM | 24 Bit |
| QRT | 24 Bit [DKB-Raytracer, QRT-Raytracer] |
| SUNRASTER | 1-8, 24 Bit (uncompressed only) |
| VLAB | internal VLab YUV-format |
| XIPAINT | 24 Bit [VDPaint, XiPaint, TruePaint] |
| JPEG | 8 or 24 Bit highly compressed |

So, if your animation is stored in one of the supported file formats: You can directly read the images with *MovieShop* - no conversion is necessary!

5.2 The JPEG Direct Import

It is very comfortable to use the multipic load, since virtually any format is supported and the whole conversion will be done automatically. But: It is relatively slow. Slow? Well, it takes a few seconds per image. If it is 24 Bit, it can easily be 1 MB large - even loading takes a while. You might think now: A few seconds - I can live with that! But: Multiply 7 seconds with 5000 images - it piles up...

So, we have also implemented a direct JPEG import. The speed is extremely high - the data will just be loaded and saved into the *VLab-Motion* partition. 15 images per second is a realistic rate!

If your animations are pretty long, you possibly save them as JPEG anyway do cope with limited harddrive space. So, if you render them to the resolution you use for *VLab-Motion* and use a playable JPEG compression, you can speed up the playback procedure a lot!

So, once again: For using the JPEG Direct import, you need to

1. Use the absolute correct resolution, e.g. 640x480

2. Store the images in the JPEG quality you use in *MovieShop*

If one of these items are not correctly set, you will get a warning and the procedure will be aborted. Review your settings in such a case! Something is obviously wrong.

While the first requirement is no major problem, the second issue is a little bit more complicated. We have used the official JFIF Q-table definitions, so if you use software like the supplied VJPEG or CJPEG (PD-software, available on multiple platforms), you just use the same percentage as you use in *MovieShop* (Define Source Window).

But other software systems like ADPro are using customized JPEG ratio definitions. ADPro, f.e., uses a scale from 1 to 1000 % !

However, it is possible to read images rendered with such programs using the JPEG Direct import. All you need to do is to convert the ADPro JPEG quality into the *MovieShop* scale.

Please refer to the read-me file on the *MovieShop* program disk for conversion tables for doing so.

Of course, if you export the images from *MovieShop* as JPEG Direct, you can directly load them back as long as the JPEG quality and resolution has not been changed.

Chapter 6

Exporting The Video Images And Digital Archiving

As you have learned by now, *MovieShop* stores the digital video on its own harddrive partition, unaccessible for other programs at this time (this maybe changed in later software versions – remember we have just started...).

Saving the video images to an AmigaOS partition can have two reasons:

- Archiving the data for doing a digital backup
- Exporting the images for rotoscoping, means for changing them with software like paint programs or image processors (ADPro, ...) and later reloading them into *MovieShop*

MovieShop is prepared for both jobs to give you effective export options.

You can save a scene or a group, so even the complete video (if you are in the root level) as one large AmigaOS file, but you can also export parts of or the complete video as single images (numbered to form a sequence).

The first method is ideal for archiving data (you can use a Streamer or Squest disks to backup the digital video), while the second one is the only suitable solution for rotoscoping.

Both methods will be explained in detail further on.

6.1 Exporting Video As A Scene

If you want to save your video just for archive or safety reasons, this is the right way to go.

It is also very useful if the related project is not finished, but you have to do a new project urgently. This would force you to clear the *MovieShop* partition, so without saving the digital video, your old project is lost.

Now how does Scene Save work?

The 'Save Scene' option from the menu takes the current scene from the Scene window, opens a file requester and saves the entire scene or group, complete with any eventually existing hierarchy of subgroups and scenes as one AmigaOS file.

If you want to save all your video in the project, just create one main group in the root level. Click on 'Root' in the Scene window, drag all the listed scenes and groups into the Clipboard window, click on 'Group' and move the new group back into the Scene window. All your video is now bundled into one main group! If you save this group, you will not miss a single frame of this project.

This file can be loaded into *MovieShop* at any time over the related 'Load Scene' command. The JPEG quality and resolution will have to stay the same, of course. After loading the scene back, you will see that the hierarchy structure is still the same – all your groups and scenes are back.

For archiving, it is a good idea to store the scene files on a removable media like a Syquest cartridge or a DAT streamer – your Amiga harddrive would fill up pretty quick otherwise.

This way of storing digital video is fast and effective – but you can only load the data back into *MovieShop*. There is no way to access the images from ADPro or any other software.

One of the coming software updates will probably feature an ADPro loader for *MovieShop* scene files, making the scene save function usable for rotoscoping as well. Right now, you will have to use the image export – coming up next.

6.2 Exporting Video As Separate Images

Very much like importing images, you can also export image data as sequences of AmigaOS files.

This time, you will have to decide in which file format you want to save the data and if you want to output fields (halfframes) or frames.

These settings need to be done in the Windows – Export Settings menu that brings up the Export Settings window.

Again, you have the choice between the fast, but unflexible JPEG Direct method and the versatile, but slower multipic method. Refer to the chapter 5 on page 38 for a brief description of these methods.

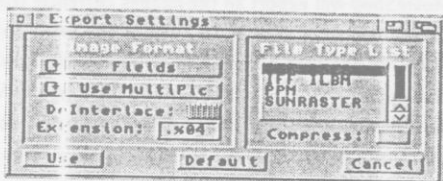


Figure 6.1: The Export Settings - Window

In short terms, once again: The multiple gives you the choice between different widely used file formats and gives you several nice options, while the JPEG Direct can only save fields rather than frames and has no further options.

Use the JPEG Direct method if you want to use software that can load JPEG images and is prepared for field rendering, like Real 3D, ADPro and more. All the options are disabled if you pick JPEG Direct.

Use the multiple method if you have software that needs specific file formats (like DeLuxePaint) and can not handle field rendering.

Also, the multiple method allows you to DeInterlace images. The DeInterlace function takes care of a very annoying effect of video processing: The movement within fields. If you have used a framegrabber like the VLab Y/C or the video toaster, you may know about this problem.

For those of you who are still confused, here's a little sidetrip into video basics:

As you might now, video is displayed using interlaced fields (60 images/sec NTSC, 50/sec PAL), two fields forming a frame. One field is using all the odd lines, the other field using all the even lines of a frame.

If the video scene contains motion like zooms or moving objects (e.g. a running man), the field images are different as well. If you export the video as frames, you would see this motion as flickering (the legs of the runner would step forth and back like mad). If displayed on a non-interlaced screen (like a VGA monitor), the effect would look comb-like.

The DeInterlace function can be used if you export frames - it will not just put the two fields together, but render the new frame. Motion within these frames will be eliminated, the output frame is nice and clean.

But: Try to keep the video in fields whenever possible. The motion looks a lot smoother if field rendering is used -- not surprising, you have 60 instead of only 30 frames per second (NTSC).

So, pick the export settings of your choice.

Now, note that *MovieShop* will export all the images currently stored in the block buffer. So, use the Block window for defining a block. If you want to export all your video, group it so you have just one root group (refer to the description above), go to the first frame, click on 'Set Start', slide to the last frame (the frame before the dummy frame) and click 'Copy' in the Block window. The buffer should now contain all your video.

Now, use the 'Export Block' function in the 'Edit' menu. The file requester asks you about base name and path – then the export starts. After the procedure is finished, you can switch to other programs for accessing the data.

You might want to give it a test run, just with like 10 images to see if you have used the correct settings.

The sequence should be numbered accurately, instantly usable with tools like FRED, the frame editor that comes with ADPro and MorphPlus. FRED is by the way well suitable for rotoscoping *VLab-Motion* video images!

Chapter 7

Macros and ARexx

7.1 General Information

This chapter outlines the macros and ARexx commands currently available with *MovieShop* version 1.0 (updates can be found on the program disk). The program is designed in a way that almost all functions and options can be controlled via ARexx commands.

The Manual *Using the System Software* contains a detailed description of ARexx. If you want to use ARexx take care that you execute the proper 'assigns', etc.

There are different ways for the user to use ARexx commands in connection with *MovieShop*:

- As individual ARexx commands:
Individual operations can be carried out through Execute Command or by using the supplied ARexx.script "MShopShell.rx" (by typing in the commands one after the other)
- As *MovieShop* macros:
Using any text editor, you can make sequences of *MovieShop* ARexx commands. These macros can then be added through Load Macro to the menu and can then be started from there. A *MovieShop* macro needs to start with #MOVIESHOP and may contain *MovieShop* commands exclusively. Compared to any normal ARexx script, *MovieShop* macros are faster and need less memory, they are, however, not as versatile.
- As Alias-Commands An Alias means to define a key that launches an ARexx command when pressed. You can put such functions on nearly any key in order to control *MovieShop* more easily.

- As ARexx scripts:

Naturally, *MovieShop* ARexx commands can be used in combination with any other ARexx commands in scripts. The latter can be executed through the menu where automatically 'RexxMast' is activated by *MovieShop* if this has not already happened. They can be opened, however, also in other ways (e.g. by double-clicking on an icon).

7.2 A Short Description of Different Parameters

The Template listing shows the type and function of any arguments that may be accepted, whether arguments are required or optional, their order, punctuation, etc. The following symbols are used:

- no / The argument may or may not be specified.
- /A The argument must Always be included.
- /N Indicates a Numeric argument, rather than a string.
- /A/N Expects Always a Numeric argument.
- /S The option simply works as a Switch.
- /M Multiple arguments are accepted.
- /K The option's Keyword must be given.

When a command name is entered followed by a single question mark, the command Template listing will be displayed.

Please note, that negative numbers in ARexx scripts must be put in inverted commas.

7.3 Return Values and Error Codes

Following are the return values and error codes that occur when commands are called via ARexx.

The following error codes exist:

- | | | | | | |
|----|---|-------|---|-----|----------------------------------------|
| RC | = | OK | = | 0: | self-explanatory |
| | | WARN | = | 5: | Warning, Cancel in File-Requester etc. |
| | | ERROR | = | 10: | Command could not be completed |
| | | FATAL | = | 20: | Wrong or missing arguments |
| | | FAIL | = | -1: | No <i>MovieShop</i> command |

If for example /S options exclude each other or none is specified where at least one has to, you will get error code 20 (FATAL). Example:

```
AUTOUSEENV           ⇒ RC=FATAL
AUTOUSEENV ON OFF   ⇒ RC=FATAL
AUTOUSEENV ON       ⇒ RC=OK
AUTOUSEENV OFF      ⇒ RC=OK
```

The ARexx variable RESULT returns values, provided that the corresponding error code is 0 (RC=OK). All possible return values are specified with the description for each command. Apart from this, you can read the current settings of the commands indicated by a * through the READ command.

7.4 The ARexx-Commands Of *MovieShop*

Command: ACTIVATEWINDOW

Template: REPLAY/S, SCENES/S, CLIP/S, MOVIELIST/S, CONTROL/S,
BLOCK/S, SOURCE/S, VIDEOPTS/S, ADVOPTS/S,
COLORKEYING/S, AUDIO/S, COMMAND/S, MACROS/S,
IMPORTOPTS/S, EXPORTOPTS/S

Purpose: This command activates a *MovieShop* window. If no parameter is specified, the main window will be activated. If the specified window is not open, an error message will be reported.

Examples: ACTIVATEWINDOW REPLAY
The RECORD/PLAY window is activated
ACTIVATEWINDOW The main window is activated.

RC: ERROR, if the window is not open.

Command: ADDMACRO

Template: FILENAME

Purpose: This command sets the name of a macro to be loaded. The loading procedure starts with the first usage of the macro. The macros added will be saved into the file "moviешop.macros" when you save the settings. You can also use the command "SAVEMACROS" to save the macros into any file.

Examples: ADDMACRO Work:Macros/MyMacro
The macro MyMacro in the folder Work:Macros will be added into the "User" menu.

ADDMACRO MovieShop:Macros/#?.ms

The ADDMACRO command supports pattern-matching. You can select multiple macros using this method.

ADDMACRO

opens a file requester to select a macro

RC: WARN, if you cancel the file requester
ERROR, if the specified macro does not exist

Command: AGC (*)

Template: SLOW/S, MEDIUM/S, FAST/S

Purpose: Control over the Automatic Gain Control

RC:

Command: ALIAS

Template: NAME/A, STRING/F

Purpose:

Examples:

RC:

Command: ASSIGNKEY

Template: KEYCODE, COMMAND=MACRONAME, REMOVEALL/S

Purpose: Putting functions on keys

Examples: ASSIGNKEY "CTRL ALT DEL" "NEWPROJECT newproject VP
DH99"

The key combination CTRL Alt Del will be assigned to a new function. A new project will be started, named "newproject". This project will use the partition DH99:. You can see that parameters that consist of several more parameters will have to be used with quotation marks.

RC:

Command: AUTORELEASEINFOSTAMPS (*)

Template: ON/S, OFF/S

Purpose: The stamps of an info window will be deleted if the window is closed. This saves memory, but the stamps will have to be recalculated if the window is opened again.

Examples: AUTORELEASEINFOSTAMPS ON

If the info window is closed, the memory previously acquired for them will be released.

RC:

Command: AUTORELEASEMOVIESTAMPS (*)

Template: ON/S, OFF/S

Purpose: Same as above, but affecting the Movie List windows instead.

Examples: AUTORELEASEMOVIESTAMPS OFF

The stamps will be kept in memory even if the Movie List gets closed. If it is opened again, the stamps will displayed immediately.

RC:

Command: AUTOSAVEENV (*)

Template: ON/S, OFF/S

Purpose: This command saves the settings (including the position and status of the windows) automatically if you quit the program.

Examples: AUTOSAVEENV OFF

The settings will not be saved when you quit. After the next start, the last saved settings will be used again.

RC:

Command: AUTOUSEENV (*)

Template: ON/S, OFF/S

Purpose: Sets if the settings will be used automatically any time you leave and restart *MovieShop*. The settings will not be saved, but kept up until you reset the computer.

Examples: AUTOUSEENV ON

If you leave *MovieShop*, the configuration will be kept in RAM and is prepared for a new program start.

RC:

Command: BANDPASS (*)

Template: I/S, II/S, III/S, IV/S

Purpose: Activates the bandpass filter.

RC:

Command: BLOCK

Template: SETSTART/S, CUT/S, COPY/S, PASTE/S, PASTEREVERSE/S,
DELETE/S, FORCE/S

Purpose: Several block operations will be performed. letzten Beenden gültig war.

Examples: BLOCK SETSTART

The current frame will be declared to be the first frame of the block.

BLOCK CUT

A block, starting with BLOCK SETSTART and ending at the current frame, will be cut out.

BLOCK COPY

A block will be copied into the internal buffer. It can be pasted again as often and to where you want. The original frames of the block will remain unchanged.

BLOCK PASTE

A previously marked block will be pasted before the current frame.

RC:

Command: BRIGHTNESS

(*)

Template: VALUE/A/N

Purpose: Changes the brightness of all color values.

Examples: BRIGHTNESS 100

Sets the brightness to 100% (the default value).

RC:

Command: BUFFERPRI

(*)

Template: PRI/A/N

Purpose: Sets the priority of the buffer task.

RC:

Command: BUFFERSIZE (*)

Template: MBYTES/A/N

Purpose: Sets the buffersize for record and play.

Examples: BUFFERSIZE 4

The buffers for both record and play will be set to 4 MB.

RC:

Command: CHROMINANCE (*)

Template: VALUE/A/N

Purpose: Sets the color saturation in percent. The value to be given has to be between 0 and 200, default is 100%.

Examples: CHROMINANCE 0

The color will be totally suppressed, effectively making in b/w.

RC:

Command: CLEARMOVIELOCATOR

Template: IN/S, OUT/S

Purpose: The given locator in the Movie List will be deleted.

Examples: CLEARMOVIELOCATOR IN

The start-locator in the Movie List window will be deleted. If the list gets played, the playback starts at the first frame and stops at either an end-locator or at the last frame of the list.

RC:

Command: CLEARSCENELOCATOR

Template: IN/S, OUT/S

Purpose: A locator that has been set in a scene is deleted.

Examples: CLEARSCENELOCATOR OUT

Deletes the end locator.

RC:

Command: CLONESCENE

Template: NAME, NEWNAME/K

Purpose: Duplicates a scene. If the parameter NEWNAME is not appended, the appendix '.dup' will be added to the original scene name.

Examples: CLONESCENE InThePool

Clones the scene InThePool, generating a new scene named InThePool.dup.

CLONESCENE Chateau Castle

The scene named Chateau will be cloned. The clone will be called Castle.

RC:

Command: CLOSEPROJECT

Template: no parameters

Purpose: The current project will be closed after the save. This is a preparation for future versions that will be able to run several projects.

RC:

Command: CLOSEWINDOW

Template: RECPLAY/S, SCENES/S, CLIP/S, MOVIELIST/S, CONTROL/S, BLOCK/S, SOURCE/S, VIDEOPTS/S, ADVOPTS/S, COLORKEYING/S, AUDIO/S, COMMAND/S, MACROS/S, IMPORTOPTS/S, EXPORTOPTS/S

Purpose: Closes the specified window. ALL will close any window currently opened. Only one window or just the ALL parameter can be specified.

Examples: CLOSEWINDOW AUDIO

The audio window is closed.

RC:

Command: COLORKEYING

(*)

Template: OFF/S, NORMAL/S, INVERSE/S

Purpose: Activates the blue box mode of the *VLab-Motion*.

Examples: COLORKEYING OFF
Deactivates the blue box module.

COLORKEYING NORMAL
The video runs in normal mode.

RC:

Command: COLORKEYU (*)

Template: LOWERLIMIT=LL/K/N, UPPERLIMIT=UL/K/N

Purpose: Sets the U values of the colors to be transparent.

Examples: COLORKEYU LL -44 UL -11
Sets U-values which, in this particular case, will make a certain red range transparent.

RC:

Command: COLORKEYV (*)

Template: LOWERLIMIT=LL/K/N, UPPERLIMIT=UL/K/N

Purpose: Same as before, but for the V-values.

Examples: COLORKEYV LOWERLIMIT 84 UPPERLIMIT 105
See above.

RC:

Command: COMMANDERRORS (*)

Template: ON/S, OFF/S

Purpose: Sets if command errors will lead to a failure requester or not.

Examples: COMMANDERRORS OFF
Even if a command is invalid, no requester comes up. An error text will be returned with RESULT.

RC:

Command: COMMANDLIST

Template: FILENAME/A

Purpose: Outputs all the ARexx commands into a file.

Examples: COMMANDLIST RAM:MovieShopCommands

A list of all the ARexx commands will be saved into the ASCII file 'RAM:MovieShopCommands'.

RC:

Command: CONTRAST

(*)

Template: VALUE/A/N

Purpose: Sets the contrast for the recorded video signal. The contrast has to be specified in percent and must be between 0 and 200%.

Examples: CONTRAST 100

The contrast will be set to the default of 100%.

RC:

Command: CORING

(*)

Template: OFF/S, LOW/S, MEDIUM/S, HIGH/S

Purpose: Sets the status of the noise filter. This enhances noisy signals.

Examples: CORING OFF

The filter will be switched off.

RC:

Command: CORRECTIMPORTHEIGHT

(*)

Template: SCALE/S, CENTER/S, REPEAT/S, CUT=FILL/S

Purpose: This command selects the procedure to be performed if an image that you want to import has the wrong height, so it does not fit to the height of VLab Motions current image settings.

Examples: CORRECTIMPORTHEIGHT SCALE

The height of the image will be scaled to a valid value.

RC:

Command: CORRECTIMPORTWIDTH (*)

Template: SCALE/S, CENTER/S, REPEAT/S, CUT=FILL/S

Purpose: Same as before, but related to the horizontal size.

Examples: CORRECTIMPORTWIDTH CENTER

The imported frames will be centered horizontally if the size is not correct.

RC:

Command: DEFAULTPUBSCREEN (*)

Template: ON/S, OFF/S

Purpose: Here, you can decide if the *MovieShop* screen will be the default public screen. Other programs will open their window directly on the *MovieShop*-screen.

Examples: DEFAULTPUBSCREEN ON

Makes the *MovieShop*-screen public.

RC:

Command: DELETEACTIVEFRAME

Template: no parameters

Purpose: The current frame will be deleted (both fields).

RC:

Command: DELETETEMOVIESCENE

Template: NAME, FORCE/S

Purpose: Deletes the specified scene in the Movie List.

Example: DELETETEMOVIESCENE Beach FORCE

The scene 'Beach' in the Movie List will be deleted without any safety requester.

RC:

Command: DELETESCENE

Template: NAME, NUMBER/K/N, NEXT/S, PREV/S, RELATIVE/N, ALL/S,
FORCE/S

Purpose: Deletes the scene specified in the parameters.

Examples: DELETESCENE TheBeach

Removes the scene 'TheBeach' from the Movie List.

DELETESCENE NUMBER 0

The first scene in the scene list will be deleted. Please note that *MovieShop* numbers the scenes starting at scene number 0. The second scene would be no. 1 and so on.

DELETESCENE PREV

The previous scene in the scene list will be deleted.

DELETESCENE 2

The overnext scene will be deleted.

RC:

Command: DEVICEPRI

(*)

Template: PRI/N, ORIGINAL/S

Purpose: Sets the priority of the harddisk device driver responsible for the *MovieShop*-partition.

Examples: DEVICEPRI ORIGINAL

Resets the device priority to the original value.

RC:

Command: DITHERSTAMPS

(*)

Template: ON/S, OFF/S

Purpose: Activates the dithering of the stamps. The stamps will look a lot better but the rendering takes longer.

Examples: DITHERSTAMPS OFF

No dithering will be used. Fast rendering with lower image quality is achieved.

RC:

Command: DOMACRO

Template: NAME/A

Purpose: Starts a MovieShop-macro. If the loaded file can not be identified as a MovieShop-macro, it will be handled due to its true nature (ARexx-script, batch-file or program).

Examples: DOMACRO MyMacro
The macro MyMacro from the user menu will be executed.

RC: RC of the last macro-command,
ERROR, if no macro exists or if the procedure was aborted.

Command: ENTERGROUP

Template: NAME, ALL/S

Purpose: A group in the scene window will be entered.

Command: EXPORTBLOCK

Template: NAME/A, FILETYPE/A, COMPRESS/S

Purpose: Export of a previously marked block in the given file format.

Examples: EXPORTBLOCK Work:OneBlock IFF COMPRESS
A marked block with the base name OneBlock will be exported in the compressed IFF format. The first image will be called OneBlock.0001, the second OneBlock.0002. If the exportsettings are set to fields, the number of files will be two times higher than the frame number in the block buffer.

RC:

Command: EXPORTCOMPRESSION (*)

Template: ON/S, OFF/S

Purpose: Activates and deactivates the compression of saved files. This is not possible with all export formats!

Command: EXPORTDEINTERLACE (*)

Template: ON/S, OFF/S

Purpose: Sets the deinterlace-function for the export of full frames. The inter-field-motion will be removed by rendering a new image.

Command: EXPORTFILETYPE (*)

Template: TYPE/A

Purpose: Specifies the file format for the images to be exported.

Command: EXPORTPICTYPE (*)

Template: FIELDS/S, FRAMES/S

Purpose: Sets if the images will be exported as frames or fields.

Command: FRAMEPOS (*)

Template: LEFTEDGE=L/K/N, TOPEDGE=T/K/N

Purpose: Sets the left and top border for the video to be recorded.

Examples: FRAMEPOS L 196 T 80

The recorded video starts at 196, 80 of the complete video signal.

RC:

Command: FRAMESIZE (*)

Template: WIDTH=W/K/N, HEIGHT=H/K/N

Purpose: Sets the frame size for the video to be recorded.

Examples: FRAMESIZE W 640 H 480

The video to be recorded is 640 by 480 pixels large.

RC:

Command: FREERUNNING (*)

Template: ON/S, OFF/S

Purpose: The VLab Motion always locks on the video signal activated in the define source window. If for some reason you are not happy with this signal, you can force the board to generate its own, broadcast quality sync. If the external signal f.e. should suffer from stability problems, you might want to use free running output.

RC:

Command: GETFRAMENUMBER

Template: NAME, NUMBER/K/N, NEXT/S, PREV/S, RELATIVE/K/N, ALL/S

Purpose: Gets the details of a frame.

RC:

Command: GETPATH

Template: TOCCATACONTROL=TC/S, PLAYSOUND=PS/S, RECORDSOUND=RS/S

Purpose: Gets the path specified for the items above.

Example: GETPATH TOCCATACONTROL

The complete path currently set in MovieShop for the ToccataControl program will be returned.

RESULT: The requested path.

Command: GETPUBSCREENNAME

Template: no parameters

Purpose: Returns the public screen name of *MovieShop*. If no own screen is used, the name of the host screen will be returned.

RESULT: Name of the public screen.

Command: GETSCENENUMBER

Template: NAME, NUMBER/K/N, NEXT/S, PREV/S, RELATIVE/K/N, ALL/S

Purpose: This command will output the number of the current scene in the scenelist. Note that *MovieShop* can identify a scene by its name or by its number, starting to count from zero.

Examples: GETSCENENUMBER JuanitasPerformance
The number of the scene 'JuanitasPerformance' will be returned as result.

GETSCENENUMBER RELATIVE 2

The number of the overnext scene (counted from the current scene) will be returned as result.

RESULT: Number of the specified scene.

Command: GETTOCCATACONTROL

Template: no parameters

Purpose: Starts *ToccataControl*.

RC:

Command: GOTOFRAME

Template: NUMBER/K/N, TIME/K, NEXT/S, PREV/S, RELATIVE/N/K

Purpose: Jumps to the specified frame in the current scene.

Examples: GOTOFRAME NUMBER 30

The frame 30 in the current scene will be the current frame.

GOTOFRAME RELATIVE -10

Jumps to the frame positioned 10 frames before the current frame.

RC:

Command: GOTOMOVIEFRAME

Template: NUMBER/K/N, TIME/K, NEXT/S, PREV/S, RELATIVE/N/K

Purpose: Jumps to the specified frame in the filmlist.

Examples: GOTOMOVIEFRAME NUMBER 0

Jumps to the first frame in the filmlist.

GOTOMOVIEFRAME TIME 0.10:10

The 10th frame in the 10th second of the filmlist will turn to the current frame.

GOTOMOVIEFRAME NEXT

Jumps to the next frame.

RC:

Command: GOTOMOVIESCENE

Template: NUMBER/K/N, NEXT/S, PREV/S, RELATIVE/K/N

Purpose: Jumps to a specified scene in the filmist.

Examples: GOTOMOVIESCENE NUMBER 0

The first scene in the filmist will be activated.

GOTOMOVIESCENE NEXT

Changes to the next scene in the filmist.

RC:

Command: GOTOSCENE

Template: NAME, NUMBER/K/N, NEXT/S, PREV/S, RELATIVE/K/N,
ALL/S

Purpose: Jumps to a specified scene in the scene list.

Examples: GOTOSCENE Triathlon

The scene Triathlon will be activated.

GOTOSCENE NUMBER 0

The first scene will be activated.

RC:

Command: GROUPSCENES

Template: SCENENAME/A/M, GROUPNAME/K

Purpose: The specified scenes will be bundled to a group. The last parameter specifies the name of the new group.

Examples: GROUPSCENES SitDown SpendFiveBucks HaveFun NumberOne

The three scenes SitDown, SpendFiveBucks and HaveFun will be bundled into the group NumberOne.

RC:

Command: HELP

Template: COMMAND/F

Purpose: Calls the help text for this command.

RC:

Command: HUE

(*)

Template: VALUE/A/N

Purpose: Changes the Hueing for an NTSC video source. Disabled for PAL sources.

RC: FATAL, if the value is beyond the legal settings.

Command: ICONIFY

Template: no parameters

Purpose: Closes the *MovieShop*-screen and puts it as an icon on the workbench screen.

RC:

Command: IMPORTFRAME

Template: NAME/A/M, UPTO/K

Purpose: Imports frames into the current scene.

Examples: IMPORTFRAME Work:gfx/Raytrace/Ball.00001 upto 250
250 images will be inserted into the current scene (Ball.00001
Ball.00250).

RC:

Command: IMPORTMETHOD

(*)

Template: MULTIPIC/S, RAWJPEG/S

Purpose: Sets the method for importing AmigaOS images. If you specify *multipic*, the file formats can be anything supported by this library (IFF, IFF Deep, SUNRASTER, PPM....). These images will then be converted into the format the Vlab Motion requires, which can take some

time. If you use RAWJPEG, the images will be imported with blazing speed, but they need to be of the right size and JPEG quality since *MovieShop* will not change them.

Command: IMPORTPICTYPE (*)

Template: FIELD/S, FRAME/S, FREDFIELDS/S, FREDFRAMES/S

Purpose: Sets the format for the images to be imported.

Examples: IMPORTTYPE FRAME

Imports images rendered as full resolution with 30/25 fps. They need to be converted into 60/50 fields per second to cover broadcast requirements. *MovieShop* does that for you.

Command: INPUT (*)

Template: YC/S, CVBS/S

Purpose: Switches between the two video inputs of the *VLab-Motion*.

Examples: INPUT YC

Uses the Y/C-input (S-VHS) for digitizing the video.

RC:

Command: INPUTVMODE (*)

Template: PAL/S, NTSC/S

Purpose: Sets the video standard.

Examples: INPUTMODE PAL

Uses PAL for recording video.

INPUTMODE NTSC

Uses NTSC for recording.

RC:

Command: INTERPOLATESTAMPS (*)

Template: ON/S, OFF/S

Purpose: Sets if interpolation should be used for rendering stamps. This raises the quality, but takes some time.

RC:

Command: IOPRI

Template: PRI/A/N

Purpose: Sets the priority for the *MovieShop* I/O-process.

RC:

Command: IRQPRI

Template: PRI/A/N

Purpose: Sets the interrupt-priority of *MovieShop* and the related libraries.

RC:

Command: JOINGROUP

Template: GROUPNAME

Purpose: The specified group will be assembled to a scene. If no groupname is specified, the current group is affected.

Examples: JOINGROUP MuscleControl

All scenes and groups inside the group MuscleControl will be melted to a scene, carrying the old group name.

RC:

Command: JPEGQUALITY

(*)

Template: VALUE/A/N

Purpose: Sets the JPEG-quality for the video compression in percent.

Examples: JPEGQUALITY 1

The digitized frames will be compressed to the minimum image quality.

JPEGQUALITY 85

The quality will be set the high-quality 85 percent.

RC:

Command: LANGUAGE (*)

Template: NAME, DEFAULT/S

Purpose: This command can be used to switch the user interface to any other supported language.

Examples: LANGUAGE deutsch
The German language will be used.

LANGUAGE DEFAULT
The workbench language will be used.

RC:

Command: LEAVEGROUP

Template: no parameters

Purpose: Leaves a group. This command is comparable with the CLI-command `tt CD /`.

RC:

Command: LOADMACROS

Template: FILENAME

Purpose: Loads a macro list.

If *MovieShop* is started, the macros in the file `ENVARC:MovieShop/MovieShop.macros` will be loaded. You can set further lists and save them with `SAVEMACROS`. This command would load such a customized list.

Examples: LOADMACROS FileName
The macro list from the file `FileName` will be used.

LOADMACROS
A file requester comes up and asks for a macro file.

RC: WARN, if cancelled in the file requester
ERROR, if the load failed.

Command: LOADSCENE

Template: NAME/A

Purpose: Loads a previously saved scene.

Examples: LOADSCENE Work:scenes/NightShift

The scene *NightShift* in the *Work:scenes* directory will be loaded.
The file name will be used as the scene name.

RC:

Command: LOADSETTINGS

Template: NAME, DEFAULT/S

Purpose: Loads the specified presetsings.

Examples: LOADSETTINGS

Opens a file requester for loading a settings file from disk or harddrive.

LOADSETTINGS DEFAULT

Die Einstellungen werden auf die internen Vorgaben zurückgesetzt.

LOADSETTINGS BigScreen

Die Einstellungen werden aus der Datei *BigScreen* gelesen.

RC: WARN bei Abbruch im Datei-Requester.

Command: LOCKGUI

Template: ON/S, OFF/S

Purpose: This command can lock or release the user interface (menues, gadgets, sliders...). It makes sense to lock the GUI while processing an ARexx script.

Examples: LOCKGUI ON

MovieShop can not be controlled with keyboard and mouse until
LOCKGUI OFF is executed from ARexx. Even terminating *MovieShop*
can only be done over ARexx.

Command: LOOP (*)

Template: ON/S, OFF/S

Purpose: Activates the loop mode for playback. You will have to use the STOP
option with the commands PLAYMOVIE or PLAY.

RC:

Command: MARKSCENEEND (*)

Template: ON/S, OFF/S

Purpose: Sets if the scene end should be marked with a black dummy frame.

Command: MOVEWINDOW

Template: REPLAY/S, SCENES/S, CLIP/S, MOVIELIST/S, CONTROL/S,
BLOCK/S, SOURCE/S, VIDEOPTS/S, ADVPTS/S,
COLORKEYING/S, AUDIO/S, COMMAND/S, MACROS/S,
IMPORTPTS/S, EXPORTPTS/S, LEFTEDGE/A/N,
TOPEGE/A/N

Purpose: Sets the position of the *MovieShop*-window. LEFTEDGE and TOPEGE specifies the x- and y-positions of the top left corner. Only one window per command can be specified.

Examples: MOVEWINDOW PLAY 200 200

The play window will be placed at the position 200,100.

MOVEWINDOW 10 10

If *MovieShop* runs on the workbench-screen, the main window will be moved to the coordinates 10,10.

RC: WARN, if the window was not closed. The new coordinates will anyway be used the next time the window will be opened.

Command: NEWPROJECT

Template: NAME, VPARTITION=VP/K, KILLDOS/S, FORCE/S

Purpose: Starts a new project.

Examples: NEWPROJECT test VP=HD0 KILLDOS FORCE

A new project will be started, named 'test'. It uses the HD0:-partition for the video data, destroying anything stored on this partition. No warning will come up. Use this command with utmost carefulness in this form!

RC:

Command: NEWSCENE

Template: NAME

Purpose: Sets up a new scene. If no scene is specified, the new scene will be named "unnamed".

RC:

Command: NUMBERTOTIME

Template: FRAMENUMBER/A/N

Purpose: Changes the number to the *MovieShop*-fileformat. This command is suitable for the conversion of a frame number generated by GET-FRAMENUMBER into the time-format.

RESULT: The time of the specified frame, using the the *MovieShop*-format minutes.seconds:frame.

Command: OPENPROJECT

Template: NAME/A

Purpose: The specified project will be opened.

Examples: OPENPROJECT Orlando94
The project Orlando94 will be opened.

RC:

Command: OPENWINDOW

(*)

Template: REPLAY/S, SCENES/S, CLIP/S, MOVIELIST/S, CONTROL/S, BLOCK/S, SOURCE/S, VIDEOPTS/S, ADVPTS/S, COLORKEYING/S, AUDIO/S, COMMAND/S, MACROS/S, IMPORTPTS/S, EXPORTPTS/S

Purpose: Opens the specified window. Only one window can be specified per command.

Examples: OPENWINDOW MOVIELIST
The movielist-window will be opened. If it was opened already, it is activated and put to front.

RC:

Command: OUTPUTVMODE (*)

Template: PAL/S, NTSC/S, AUTO/S

Purpose: Sets the video standard for playback/display.

Examples: OUTPUTMODE AUTO

The video will always be played in the standard used for record.

RC:

Command: PAUSE (*)

Template: ON/S, OFF/S, TOGGLE/S

Purpose: Activates, deactivates or toggles the PAUSE.

Examples: PAUSE TOGGLE

Inverts the condition of the PAUSE gadget.

RC:

Command: PLAY

Template: DOWAIT/S

Purpose: Plays the current scene or movielist. DOWAIT will let the ARexx script stand by until the Play is done.

RC:

Command: PLAYBURSTSIZE (*)

Template: KBYTES/A/N

Purpose: Sets the blocksize for play.

Examples: PLAYBURSTSIZE 256

The blocksize used for playback is 256 kByte. *MovieShop* requests the data in 256 kByte pieces from the harddrive device driver.

RC:

Command: PLAYSOUND (*)

Template: ON/S, OFF/S

Purpose: Sets is a sound file should be played with the video.

RC:

Command: PREFILTER (*)

Template: ON/S, OFF/S

Purpose: Sets the prefilter for the CVBS input.

RC:

Command: PRELOADSIZE (*)

Template: KBYTES/A/N

Purpose: Sets the amount of data that is preloaded for playback. The preload size needs to be smaller than the buffer size indeed.

Examples: PRELOADSIZE 1024

1 MB data will be preloaded before the playback starts, so 1 MB reserve will be available right from the start.

RC:

Command: PROCEED (*)

Template: SCENES/S, MOVIELIST/S

Purpose: Toggles between scene and movielist mode.

Examples:

RC:

Command: PROJECTVMODE

Template: PAL/S, NTSC/S

Purpose: Sets the video standard for the project.

Command: QUIT

Purpose: Sets the length of a scene to be recorded.

Examples: RECORDLENGTH UB

The record procedure will be continued until STOP is used, either from ARexx or from the GUI.

RECORDLENGTH SINGLEFRAME

Exactly one frame gets recorded.

RECORDLENGTH FRAMES 15:00:00

One scene with 15 minutes length will be recorded.

RC:

Command: RECORDSOUND

Template: ON/S, OFF/S

Purpose: Sets if audio data (from the Toccat board) will be recorded with the video.

RC:

Command: RELEASESTAMPS

Template: INFO/S, MOVIELIST/S

Purpose: Releases the memory occupied by the stamps. If stamps are still open, this will not be closed. If they are open, but overlaid with other windows, they will be refreshed after the get visible again.

Examples:

RC:

Command: REMOVEMACRO

Template: NAME/M, ALL/S, FORCE/S

Purpose: Removes one or all macros from the user menu. FORCE suppresses any safety requester.

Examples: REMOVEMACRO AutoRecord

The macro AutoRecord will be deleted in memory.

RC: WARN, if a safety requester has not been confirmed.
ERROR, if a macro can not be removed because it is currently performed or unavailable.

Command: RENAMESCENE

Template: NEWNAME/A

Purpose: Renames the current scene in the scene list.

Examples: RENAMESCENE HoundDog
The current scene will be renamed as HoundDog.

RC:

Command: RESET

Template: no parameters

Purpose: Resets the chip sets on the *VLab-Motion*.

Command: SAFETYLEVEL (*)

Template: LOW/S, MEDIUM/S, HIGH/S

Purpose: Sets how many safety requesters will come up.

Examples: SAFETYLEVEL LOW
Even at QUIT or DELETESCENE, no safety requester will appear.
SAFETYLEVEL MEDIUM
Only serious actions will lead to a requester.
SAFETYLEVEL HIGH
Any unreversible action will lead to a safety requester.

Command: SAVEICONS (*)

Template: ON/S, OFF/S

Purpose: Sets if *MovieShop*-files will be accomplished with an icon or not.

Examples: SAVEICONS OFF
No icons will be saved with the *MovieShop*-files.

Command: SAVEMACROS

Template: FILENAME

Purpose: Saves the macro list. If "Auto-Save" is activated, the macro list ENVARC:MovieShop/MovieShop.macros will be saved automatically when the program is terminated.

Examples: SAVEMACROS FileName
The macro list will be saved as FileName.

SAVEMACROS

A file requester appears and lets you specify a path and name for the macro file.

RC: WARN, if cancelled in the safety requester
ERROR, failure at writing

Command: SAVEPROJECT

Template: no parameters

Purpose: The current project will be saved.

RC:

Command: SAVEPROJECTAS

Template: NAME/A, FORCE/S

Purpose: Saves the project using the specified name. If FORCE is used, any eventually existing file with this path and name will be immediately overwritten.

Examples: SAVEPROJECTAS Holiday95
The current project will be saved as 'Holiday95'. If the file exists already, a requester will appear.

RC:

Command: SAVESCENEAS

Template: NAME/A, FORCE/S

Purpose: The current scene will be saved under the specified name. FORCE overwrites existing files.

Examples: SAVESCENEAS Work:MyScene
The current scene will be saved as MyScene on the device Work:. If such a file exists already, you will get warned.

RC:

Command: SAVESETTINGS

Template: NAME

Purpose: Saves the current settings of *MovieShop*.

Examples: SAVESETTINGS

A file requester appears, asking for path and filename of the settings file to be saved.

SAVESETTINGS ENVARC:MovieShop/MovieShop.prefs

The settings will be saved as ENVARC:MovieShop in the file *MovieShop.prefs*. This file will be used by default.

RC: WARN, if cancelled in the file requester.
ERROR if the save procedure failed.

Command: SCENEINFO

Template: no parameters

Purpose: Opens an info window for the current scene.

RC:

Command: SCENETOMOVIE

Template: APPEND/S

Purpose: Copies the current scene from the scene list into the movie list. If APPEND is specified, the scene is put after the current position, else before.

RC:

Command: SCREENDDEPTH

(*)

Template: DEPTH/A/N

Purpose: Sets the number of bitplanes for the *MovieShop*-screen.

Examples: Screendepth 4

A screen with four bitplanes (16 colors) will be opened.

RC: FATAL if DEPTH higher than 8.

Command: SCREENFONT (*)

Template: NAME, SIZE/N, DEFAULT/S

Purpose: Sets the screen font for the title bar text of the windows.

Examples: SCREENFONT helvetica 15

The font 'Helvetica 15' will be used.

SCREENFONT

A font-requester will be opened.

SCREENFONT DEFAULT

The font selected by the preferences program 'font' will be used.

RC: WARN if cancelled in the font requester.

Command: SCREENTOBACK

Template: no parameters

Purpose: Puts the *MovieShop*-screen to background.

RC: ERROR, if the screen was not open.

Command: SCREENTOFRONT

Template: no parameters

Purpose: Brings the *MovieShop*-screen to front.

RC: ERROR, if the screen was not open.

Command: SELECTPUBSCREEN

Template: NAME, NEW/S, DEFAULT/S

Purpose: Selects the public screen that *MovieShop* will use.

Examples: `SELECTPUBSCREEN CygnusEdScreen1`
MovieShop opens all of its windows on the screen named 'CygnusESS-screen1'.

`SELECTPUBSCREEN NEW`
A new public screen will be opened for *MovieShop*.

RC: WARN, if cancelled in the file requester. bei Abbruch im Requester.

Command: `SETPATH`

Template: `TOCCATACONTROL=TC/K, PLAYSOUND=PS/K,
RECORDSOUND=RS/K`

Purpose: Sets the path for the audio files that *MovieShop* will record and play with the video and for the *ToccataControl* program.

Examples: `SETPATH TC Work:ToccataTools/Toccatacontrol`
ToccataControl can be found under its name in this directory.

RC:

Command: `SETREFFRAME`

Template: `TIME`

Purpose: Sets a reference frame for the current scene in the scene list. If no parameters are specified, the current frame will be the reference frame further on.

Examples: `SETREFFRAME time`
The frame at the specified time will be the reference frame. The time needs to be in `minute.second:frame` format.

RC:

Command: `SHOWFRAMEMODE`

(*)

Template: `REFERENCE/S, FIRST/S, EVEN/S, ODD/S`

Purpose: Sets the display mode.

Examples: `SHOWFRAMEMODE EVEN`
Still images will be displayed with their even fields on the connected TV monitor.

SHOWFRAMEMODE REFERENCE

At the setting of a scene, the reference frame will be displayed rather than the first frame. If no reference frame is defined, the first frame will be displayed however.

RC:

Command: SHOWINFOSTAMPS (*)

Template: REFERENCE/S, ALL/S

Purpose: Sets the format of the info window for displaying stamps.

Examples: SHOWINFOSTAMPS REFERENCE

Shows a stamp size reproduction of the reference frame in the info window for this scene or group.

SHOWINFOSTAMPS ALL

The info window for a scene consists of the first frame, the reference frame and the last frame for the scene. If no reference frame exists, the middle field remains empty.

RC:

Command: SHOWMOVIESTAMPS (*)

Template: NON/S, EXISTS/S, ALL/S

Purpose: This command sets which stamps should be displayed in the movie list window.

Examples: SHOWMOVIESTAMPS NON

No stamps will be displayed in the movielist.

RC:

Command: SMOOTHING (*)

Template: ON/S, OFF/S

Purpose: Activates and deactivates the smoothing-filter.

RC:

Command: SPLITSCENE

Template: TIME, NAME1/K, NAME2/K

Purpose: Splits the scene at the specified point.

Examples: SPLITSCENE 0.30:00 FirstPart SecondPart

The current scene from the scene list will be split at second 30 (frame 900 (NTSC)). Two new scenes are generated, FirstPart (30 secs) and SecondPart (the remaining).

RC:

Command: STOP

Template: no parameters

Purpose: Stops record and playback.

Command: SUBCARRIER

(*)

Template: VALUE/A/N

Purpose: Sets the frequency of the subcarrier signal.

Command: TIMETONUMBER

Template: TIMESTRING/A

Purpose: Converts a time in the format minutes.seconds:frames into a frame number, starting from zero onwards.

Examples: TIMESTRING 10.00:10

RESULT gives the number 18010 (at 30 fps NTSC) back.

RC:

RESULT: A number representing the frame number at the specified time.

Command: UNALIAS

Template: NAME/A

Purpose: Removes a command defined with ALIAS.

Command: UNGROUPSCENE

Template: NAME

Purpose: A group will be disassembled. Any scene or group previously bundled into it will show up again.

Examples: UNGROUPSCENE

The current group in the scene list will be disassembled.

UNGROUPSCENE XMasEve

The group XMasEve will be disassembled.

RC:

Command: UNZOOMWINDOW

Template: REPLAY/S, SCENES/S, CLIP/S, MOVIELIST/S, CONTROL/S,
BLOCK/S, SOURCE/S, VIDEOPTS/S, ADVPTS/S,
COLORKEYING/S, AUDIO/S, COMMAND/S, MACROS/S,
IMPORTPTS/S, EXPORTPTS/S

Purpose: A zoomed window will be brought back to full size.

Examples: UNZOOMWINDOW RECORD

The record window will be brought back to full size after it was iconified (zoomed).

RC:

Command: VERSION

Template: VERSION/N, REVISION/N, FULL/S

Purpose: Checks the version- and revisionnumber. If a version number is specified, no versionstring will be returned.

Examples: VERSION

Get the versionstring without the date.

VERSION FULL

Get the entire versionstring.

VERSION 6 2

Check if the version is at least 6. If it is higher, RC is zero, if it is lower, the RC (Warning). If it is just the same, it will further be checked if the revision number is at least 2.

RC: WARN, if version or revision is to low.

RESULT: Versionstring, if RC=OK.

Command: VNR

(*)

Template: NORMAL/S, WINDOW/S, FREE/S, OFF/S

Purpose: Activates the mode for the Vertical Noise Reduction. Several filters will be used for eliminating disturbances in the VSync-signal.

RC:

Command: VTR

(*)

Template: ON/S, OFF/S

Purpose: If a video recorder is used as a source, the "Time Base Correction" should be activated using this command.

Examples: VTR ON

The "Time-Base-Correction" is activated.

Command: WEIGHT

(*)

Template: OFF/S, WEAK/S, MEDIUM/S, STRONG/S

Purpose: Sets the weight on the filters 'Prefilter' and 'Smoothing'.

RC:

Command: WINDOWFONT

(*)

Template: NAME, SIZE/N, DEFAULT/S

Purpose: Sets the font to be used in windows. wird.

Examples: WINDOWFONT Topaz 8

The font 'Topaz 8' will be used for *MovieShop*-windowtexts.

WINDOWFONT DEFAULT

The configured font from the preferences-font-program will be used in windows.

WINDOWFONT

A font requester will be opened and lets you select a font.

RC: WARN if cancelled in the font requester.

Command: WINDOWTOBACK

Template: RECPLAY/S, SCENES/S, CLIP/S, MOVIELIST/S, CONTROL/S,
BLOCK/S, SOURCE/S, VIDEOPTS/S, ADVPTS/S,
COLORKEYING/S, AUDIO/S, COMMAND/S, MACROS/S,
IMPORTOPTS/S, EXPORTOPTS/S

Purpose: A window will be brought to back.

RC: ERROR, if the window is not open
FATAL, if no window was specified.

Command: WINDOWTOFRONT

Template: RECPLAY/S, SCENES/S, CLIP/S, MOVIELIST/S, CONTROL/S,
BLOCK/S, SOURCE/S, VIDEOPTS/S, ADVPTS/S,
COLORKEYING/S, AUDIO/S, COMMAND/S, MACROS/S,
IMPORTOPTS/S, EXPORTOPTS/S

Purpose: The specified window will be put to front.

RC: ERROR, if the window was closed
FATAL, if no window was specified.

Command: WORKBENCH (*)

Template: OPEN/S, CLOSE/S

Purpose: Opens or closes the workbench screen.

Examples: WORKBENCH CLOSE

Tries to close down the workbench screen. Returns an error code if
the workbench can not be closed for some reason.

RC: ERROR, if the screen could not be closed.

Command: ZOOM

Template: OUT/S, IN/S, MIN/S, MAX/S

Purpose: Sets the zoom factor in the movielist.

Examples: ZOOM MAX
The maximum zoom factor for the movielist will be used.
ZOOM OUT
The zoom factor is lowered by one level.

RC:

Command: ZOOMWINDOW

Template: RECPLAY/S, SCENES/S, CLIP/S, MOVIELIST/S, CONTROL/S,
BLOCK/S, SOURCE/S, VIDEOPTS/S, ADVPTS/S,
COLORKEYING/S, AUDIO/S, COMMAND/S, MACROS/S,
IMPORTPTS/S, EXPORTPTS/S

Purpose: The specified windows will be brought to minimal size. Only the title bar remains visible.

RC: ERROR, if the specified window was closed
FATAL, if no window was specified.

7.5 ARexx Application Example: Using *VLab-Motion* With Scala MM

The reputation of the Amiga as the optimal video computer is certainly based on its outstanding hardware and operating system, but without exceptional software packages, noone would have been able to use these features in real life.

Examples of such products are certainly the already mentioned ADPro and MorphPlus from ASDG/Elastic Reality, but at least as important is the amazing Scala MultiMedia 300, a software system for video work and presentation. Of course, the high end InfoChannel version does the same job for professional applications.

One of the most often reported question regarding the *VLab-Motion* is: So, can I use it from Scala?

The answer is: Yes, you can. Due to the powerful ARexx ports of *MovieShop* and Scala, you can integrate *VLab-Motion* video clips with audio (using the Toccata board) in one presentation, even in a one-monitor-setup!

As a Scala user, you know that Scala is based on events. Such events can be text effects, audio and more - even external devices can be controlled over so-called EX modules. An EX module would normally be the right way to go. We have supplied a *VLab-Motion* board to Scala for development purposes, but its early days...

Anyway, another possible Scala event is an ARexx script launch. That's all we need! The *MovieShop* system can be controlled entirely over ARexx.

The procedure to play *MovieShop* video/audio scenes over Scala is:

1. Set a Scala background picture with buttons like you always do in Scala. Reserve one button for playing a video/audio clip.
2. Save the Scala script and load it into a text editor (CygnusED, MicroEmacs, ED, ...). Now, add the following text to the original script:

```
EVENT "Clip 1"  
EXECUTE s:SVMTest.rx arexx  
GOTO "Main [the name of your main menu]"  
END
```

This command sequence will execute the to-be-defined ARexx-script 's:SVM-Test.rx', launched by a Scala button-click.

3. Save the script, load it back into Scala (replace the original one).
4. Now connect your reserved button to the new event (button menu).
5. Now, start your editor again and define the following text:

```
/* Test for Arexx launched Sceneplay  
*/
```

```
ADDRESS MOVIESHOP
```

```
GOTOSCENE NUMBER 0
```

```
PLAY DOWAIT
```

```
STOP
```

```
EXIT
```

6. Save this little text file as 'S:SVMTest.rx'.
7. Start *MovieShop*. Make sure you have at least one scene or group in the Scene window.
8. Start your Scala script and click on the reserved button. The video/audio should play by now!

You can easily play more than one scene, just add another event and another script. You could even expand your Amiga into an all-digital video juke box, controlled by Scala!

Now, we mentioned that you can use a *VLab-Motion*-equipped Amiga as a one-monitor Scala-system integrating *VLab-Motion* digital video. Well, it is easier than it sounds! Just take a video encoder (available from us, but also integrated into most Genlocks) and connect it to the Amiga RGB video port.

Any Amiga video will now be coded into Y/C or composite video, suitable for feeded into the *VLab Motion*.

Assuming the clips for *MovieShop* have been prepared before, the *VLab-Motion* now locks on the Amiga video. If it plays the prepared clip, the Amiga video is shut down and will reappear after the scene playback is finished.

You need to display the Scala interface that you have picked in interlaced video for this genlock-supported feature.

So, the user sees his Scala background on the TV screen (Amiga signal put through the *VLab-Motion*). If a gadget with video playback is started, the video turns into the scene and returns to Scala after finished playback.

If the screen turns black when playing or just displaying video: Switch the 'Free Running' gadget in the Video Options window off. You need to genlock on the Amiga video to display both video signals on one monitor.

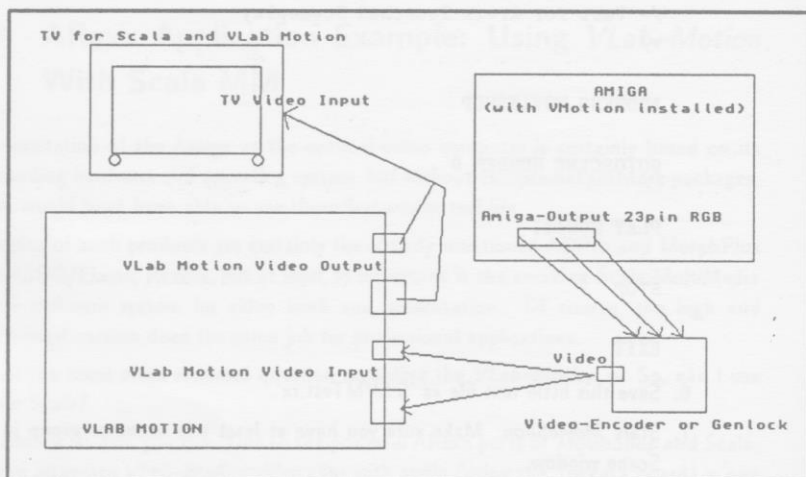


Figure 7.1: Hardware Concept for a One-Monitor-System

Chapter 8

Future Plans For *MovieShop*

If you have purchased MacroSystem products in the past, you certainly know that we invest lots of effort in the enhancement of our system software. At the time these words are written, the VLab software is at 4.2, and the Retina Display Board software is at 2.2.

MovieShop and *VLab-Motion* is by far the most complex, but also the most exciting product we have done so far. It will probably see many software updates – we have lots of ideas even right now.

At this point, we would like to tell you about our future plans. If you need anything else or disagree with our priorities, let us know! We are not too arrogant to ignore the feedback of you, our customer.

Also, many third party companies have applied for samples of *VLab-Motion*. We have supplied boards to all serious volunteers! So, expect third party products with *VLab-Motion* support in the near future. If you are a potential developer: Contact us for the documentation of the `vmotion.library` and the `motioneffect.library`!

Now, find a list of planned features following! We will not talk about ideas – this would fill an own book easily. The things listed here are concrete and secure. Watch out!

8.1 Digital Effects

As already mentioned, you can use any of the outstanding image processors to create digital effects, just by exporting and re-importing images from *MovieShop*.

The results are certainly astounding, but we realized that this method is to time-intensive and complicated. So, we have decided to invent effects in *MovieShop*! Things like fades, wipes, shifts, negative, rotate and many more are possible, only requiring 3 - 5 seconds per image rendering time.

For this purpose, the next version of the time line will be horizontal and has room for more than one scene at a time. You can overlay two scenes, like the last second of the first with the first second of the second scene. For these 30 frames, you then need to define an edit event, like a fadethrough or a wipe. Just take the effect from the Effect List (a new window) and drag it into the timeline, into the one-second-overlay time slot.

The positioning works with *MovieShop*'s common drag-and-drop method.

You can also define effects for a single scene, like negative or mosaic. Just drop the effect parallel to the scene to be affected and stretch the effect over the targetted time!

If you have finished, you can execute the rendering. *MovieShop* steps through the time line and renders effect frames if it finds unrendered effect settings. The output image is displayed on the TV monitor immediately after rendering for on-line control.

This brings another powerful feature into *MovieShop*: It turns the Amiga into a digital video effect generator.

By the time these words are written, the new version is in late Beta and was already presented to the public at the SIGGRAPH 94 show in Orlando, Florida. You can expect it very soon (in fact, check your disk for readme-files – maybe you already got *MovieShop* 1.2).

Since the effects are modular and expandable just by copying new effect files into the effect drawer, you can further expect more exciting operators – even from third party due to the open developer support program of *MovieShop*.

8.2 Audio Editing

After we have finished 1.2 with the effects, we plan to focus on the audio side.

Our current plan is to setup a second dedicated partition for the audio (can be much smaller than the video partition). This enables us to edit audio and video together, without the slowness of the AmigaOS filesystem. The same non-destructive editing will guarantee seamless audio.

The audio can even follow video edit events like fadethroughs!

The time line will most certainly be responsible for the audio side then.

8.3 *MovieShop* Pro And Hardware Modules

The *VLab-Motion* is a device done for the "prosumer" market. It is an online system for users working with VHS, S-VHS and Hi8-equipment.

Those users will find all necessary features in the 'regular' package.

User feedback has brought to daylight that, especially in the USA, many professionals on tight budgets plan to use the *VLab-Motion*, too.

Those people have different requirements. They need things like

- YUV component video output to interface with Betacam/SP
- Off-Line editing for non-compressed, Betacam quality (using the *VLab-Motion* as an Edit Decision List generator for controlling High-End analog editing machinery)
- Direct hardware control like RS232 and GPI

We have decided to come with a software version of *MovieShop* for these users. Also, we are prototyping the YUV component adaptor that connects to the *VLab-Motion* expansion socket, available on any *VLab-Motion*. A RGB version will follow.

Due to the smaller market potential and higher support efforts, we will charge extra for both software and hardware. Call for being put on the list of interested users!

Chapter 9

Troubleshooting

The VLab Motion Record and Playback is a procedure that runs at extremely high performance levels. It may happen that you have problems with your system.

We want to give you some hints that should tell you what to do in just such a situation.

Problem: The system is not playing back and recording at high data rates

Try to lower or raise pre-load size and block sizes in the Define Source - Advanced Options window. Some hard disk controllers like the A 4091 and the Fastlane Z3 seem to work better using small pre-read and block sizes.

Problem: The video lead through is messed up

Check your cabling! Have you selected the right video standard (NTSC or PAL?)

Problem: No video appears on the output

This may sound strange: Have you activated the right input? If no signal is detected, no signal can be genlocked upon.

Problem: The video gets black during playback, the system sometimes even hangs afterwards

If you bring your hardware to its limits, it can happen that the JPEG chip set gets confused. Try to reduce the quality level. Also check your offsets and resolutions. If this does not help, try to remove all other installed Zorro boards and see if one of them causes the problem.

Problem: The video is disturbed when digitizing from video tape

Have you activated the "VTR" switch in the "Define Source" window?

Problem: The video flickers in the color parts

Try to change the Sub Carrier settings in the Define Source - Video Options window.

Problem: The video flickers strongly when scrolling through it with the cursor keys or the Control window slider

That is a software limitation of the early versions of the software. Call for an update!

Problem: When using the chroma keying, any time I play the video, my camera turns black. Still images do work.

This is a typical problem. You certainly have checked the 'Free Running' option in the Video Options window. The card will then generate its own high-quality video signal – but the camera displaying the blue box object is not usable. Deactivate the free running option and that's it!

Problem: The video output is too high. A black border appears at the bottom, and the image can not be positioned any lower using the vertical offset settings.

The very first boards shipped had such problems due to a wrong sync. Call for getting your board fixed (for free).

Problem: The first record after bootup often fails. An "internal error 19" happens. Afterwards, the system works just fine.

This is normal. To completely initialize the chip set, it is necessary to perform a record. You may want to do that any time you start MovieShop.

Problem: When I load my rendered animation, MovieShop refuses to play it. It would not even display a single still!

You will probably have to feed in a reference NTSC or PAL signal, at least one time after cold start to initialize the VLab-Motion hardware. This problem will be solved in one of the next software updates.

PROBLEM: I am missing my 200 jokes! Come on, I want them! What do I need to do?

Well, this is a tough issue. Obviously, our Moral Standards Statistical Science Group (MSSSG) has decided to put your country into the pool of anti-sexual nations. Sorry for you! But: You can take action. Write to your president for more liberalism – send us his answer! Even more important: Change your personal behaviour, also to set marks for others. Visit striptease bars at least once a week, unfold the Playboy centerfold in public (e.g. at your haircutter), subscribe to at least two major nudistic magazines! We review each country once a year, so you maybe able to get hold of the famous VLab-Motion jokebook soon.

HUGE PROBLEM: My feet are extremely cold

Remember we told you we were going to knock your socks off? Try wearing heavy boots, preferably with strong laces. A heating pad may also prove useful.

Appendix A

Literature List

For those of you being interested in JPEG and image compression, this list might be of interest.¹

As you see, some of the articles and books are German - one more reason for you to start taking courses...

- Nick Baran. *Bilderpresse – Datenkompression in der Bildverarbeitung*. C't Magazin für Computertechnik, Page 44-49, Februar 1991.
- J. Biemond und E.D. Frimout *Video Compression: Techniques and Standardisation Activities*. IEEE Proceedings Multimedia, 1992.
- Claas Burghard. *Bildverarbeitung/Bilddatenkompression*, Hauptseminar am Institut für Informatik, Clausthal, 1991.
- Didier le Gall. *MPEG: A Video Compression Standard for Multimedia Application*. Communications of the ACM, April 1991, Vol.34, No.4.
- Peter Haberäcker. *Digitale Bildverarbeitung – Grundlagen und Anwendungen*. Carl Hanser Verlag, 2. Auflage München 1987.
- Josef Hoffmann. Redundanz raus – Bildkompression mit DCT und anderen Transformationen. C't Magazin für Computertechnik, Page 126-128, Juni 1991.
- G. Hudson, et. al. *The International Standardisation of a Still Picture Compression Technique*, Globecom 1988.
- ISO/IEC JTC1/SC2/WG11. *Committee Draft of Standard ISO11172, Coding of moving pictures and associated audio*, MPEG 90/176 rev.2ed, Dec 1990.

¹This list is a selection from the diploma thesis "Parallelisierung des Bildkompressionsstandards JPEG" (Parallelization Of The Image Compression Standard JPEG), which has been written at the "Institut für Informatik der TU Clausthal" by Dipl.-Ing. Guido Falkenmeyer

- ISO/IEC JTC1. *Committee Draft of Standard ISO 10918-1, Digital Compression and Coding of Continuous Still Images, Part 1, Requirements and Guidelines*, ANSI, Februar 1992.
- ISO/IEC JTC1. *Committee Draft of Standard ISO 10918-2, Digital Compression and Coding of Continuous Still Images, Part 2, Compliance testing*, ANSI, März 1993.
- Oliver Kesy. *Bilder schrumpfen – Neue Methoden und Programme zur Bildkompression*. C't Magazin für Computertechnik, Page 120-126, November 1993.
- A. Léger, et. al., *Still Picture Compression Algorithms Evaluated for International Standardisation*, Globecom 1988.
- Gregory K. Wallace. *The JPEG Still Compression Standard*. Communication of the ACM, April 1991, Vol34, No.4.

Appendix B

Report

If you detect errors or bugs, we want and need your report. To be able to reproduce your failures, we also need to know details about your system. So, please copy this page and fill it in, then fax it or mail it to us. Your help is highly appreciated.

Enhancement Suggestions

Please cross:

Failure Report

Program: MovieShop

Version : _____

Used Computer: _____

Processor / Freq.: _____

Memory: _____ MB ChipMem, _____ MB FastMem

AmigaOS Version: _____ WB: _____ Kickstart: _____

Harddrive(s): _____

Harddrive Controller: _____

Additional Hardware: _____

""Utilities"" in Background: _____

Error Report / _____

Suggestion: _____

Appendix C

Hotkeys

Many functions of *MovieShop* can be launched over keyboard hotkeys as well. Since the amount of available hotkeys exceeds the number of menu functions, here is a list for your reference.

C.1 Menu-Shortcuts

These shortcuts can be launched by selecting the menu point or by pressing the right Amiga key plus the listed key together.

C.2 Controlling *MovieShop* With The Keyboard

As you know, you can "walk" through your video using the Control and the Scene windows. To help the customer who likes to use the keyboard for faster access, we have defined keyboard equivalents for these windows. The following table shows the function, the key and the related ARexx command.

MacroSystem

| Menu Point | | Key |
|--------------------|----------------------------------------|------------------|
| Project: | >> New... | N |
| | >> Open... | O |
| | >> Save | S |
| | >> Save as... | A |
| | >> Projekt Information... | ? |
| | >> Quit MovieShop | Q |
| Edit: | >> Import Frames >> Import | I |
| | >> Delete Frame | K |
| | >> Mark Block | B |
| | >> Cut | X |
| | >> Copy | C |
| | >> Paste | V |
| | >> Delete | D |
| | Windows: | >> Define Source |
| >> Record/Play | | 2 |
| >> Scenes | | 3 |
| >> Movie List | | 4 |
| >> Control | | 5 |
| >> Block | | 6 |
| >> Clipboard | | 7 |
| >> Audio | | 8 |
| >> Execute Command | | E |
| >> Close Window | | 0 |
| Settings: | >> Screen-Type >> Select Pub-Screen... | J |

Table C.1: Menue-Shortcuts

| Key | Function | Alias |
|--------------------|-------------------------|-------------------------------------|
| Esc | Quit Record | STOP |
| Esc | Quit Play | STOP |
| Return/Enter | Enter Group | ENTERGROUP |
| Backspace | Leave Group | LEAVEGROUP |
| Cursor right | Next Frame | GOTOFRAME NEXT |
| Cursor left | Previous Frame | GOTOFRAME PREV |
| Cursor up | Previous Scene | GOTOSCENE PREV |
| Cursor down | Next Scene | GOTOSCENE NEXT |
| Alt-Cursor right | Previous Scene | GOTOSCENE PREV |
| Alt-Cursor left | Next Scene | GOTOSCENE NEXT |
| Ctrl-Cursor right | Scene End | — |
| Ctrl-Cursor left | Scene Start | GOTOFRAME NUMBER 0 |
| Shift-Cursor right | 10 Frames Forth | GOTOFRAME RELATIVE 10 |
| Shift-Cursor left | 10 Frames Back | GOTOFRAME RELATIVE -10 |
| Space | Toggle The Control Mode | PROCEED SCENES PROCEED MOVIELIST |

Table C.2: Controlling *MovieShop* over the keyboard