

## 5. Introduction To Jaleo Software Configuration

Jaleo software configuration is performed with a number of configuration files. These files are text files that can be edited with any text editor. All configuration files are contained in the `JALEO-ENV` directory that is placed in the home directory of all login accounts prepared to use Jaleo. For information on how to prepare an account for Jaleo, see “Using Multiple Accounts” on page 35.

### 5.1 Configuration Warning

That all user configurable files are placed in the `JALEO-ENV` subdirectory in your home directory does **BY NO MEANS** mean that all files in that folder can be edited. Only files described in this manual should ever be edited by the end user. You may jeopardize the proper operation of your Jaleo system by not conforming to this rule - you have been warned.

Whenever you intend to change a configuration file, please make a copy of the file before you actually change something. This will allow you to go back to a previous state easily in case you make a mistake.

The configurable files are these:

- The main configuration file `.jaleorc` in `JALEO-ENV`
- The work directory configuration file `JALEO_WORK` in `JALEO-ENV`
- The device and configuration files in `JALEO-ENV/etc/devices`. Configurable are files with the extension `.dev` and `.cfg`.
- The script files `input_process` and `output_process` in the `JALEO-ENV/utils` directory (see the release notes for more information).

No other files should be touched by the user.

### 5.2 Editing Configuration Files

The tool of choice for editing configuration files is the SGI text editor `jot`. We limit our discussion here to this editor; if you are thinking about using other editors like `vi` or `emacs`, you are probably an experienced UNIX user anyway, so that this discussion is not of great interest to you.

#### 5.2.1 Using the Graphical Desktop

The easiest way to access files for editing is the SGI desktop. You can find `jot` in the Icon Catalogue, category “Applications”. Open the Icon Catalogue window by selecting the respective submenu entry in the Find toolchest menu. If you want to have more immediate access to the editor, you can drag the icon to the desktop. Click and hold the left mouse button on the `jot` icon while you move it to any desktop location you like.

To edit a configuration file using `jot`, you can simply drag its icon from a directory view onto the `jot` icon. The directory view icon of your home directory is (in a default SGI account) visible as a folder icon somewhere on the very right side of the screen. The folder is named after your login name.

You can open the folder by double clicking on it. Also open the JALEO-ENV directory (that should be visible inside the directory view of you home directory, provided your home account is set up for use with Jaleo).

Just for practise, let us bring the configuration file JALEO\_WORK in the editor. To do so, drag the icon over the editor icon (the editor icon will change its colour to blue when you are over it) and let the mouse button go. An editor window will open that shows the content of the file, that is only a single line. Quit the editor by selecting `Exit` from its File menu.

Explaining further operation of the desktop is beyond the scope of this manual - please see the SGI documentation that came with your system for further information. Also you can use the online desktop help system from the Help menu of the system toolchest.

### 5.2.2 Using a Shell

To use a shell for configuration file access, first open a shell window from the System menu of the system toolchest. You can only type in shell windows while the mouse cursor is positioned over the window.

All UNIX commands that you enter into the shell need to be executed by pressing return.

The most important commands we will use are briefly explained below. Please see the man pages for more information:

- `cd`

The `cd` command is used to change the current directory. `cd` without any arguments changes to you home directory, if you want to go somewhere specific you have to enter a directory pathname. All UNIX pathnames consist of directory names separated by slash characters `“/”`.

- `pwd`

Prints out the current directory, i.e. the directory you currently are positioned in.

- `ls`

The `ls` command gives a list of the current directory content. There are useful options to this command, and one shall be mentioned here:

- `ls -l` gives a listing with a lot more information about each file

- `man`

Using the `man` command, you can get more information about any other UNIX command. Just type:

`man <command name>`

For example:

`man ls`

- `df`

Gives you the total amount of disk space on each mounted file systems, as well as used and free disk space. By default, the values printed are in disk blocks; to receive values in KBytes, please type

`df -k<return>`

To make sure you are at your current directory, please type

```
cd<return>
```

<return> means: press the enter or return key, do not type the letters. Now change to the directory JALEO-ENV

```
cd JALEO-ENV<return>
```

To see the content of the directory, type

```
ls<return>
```

To edit a file, type the name of the editor and the name of the file you want to edit, for example

```
jot JALEO_WORK
```

A common mistake while editing configuration files is to forget a terminating line feed at the end of the last line of a configuration file. If you do forget the last line feed, the last line will be ignored. The SGI text editor, `jot`, does automatically warn you if you try to save a file without a new line character at the end. This is a very convenient method to prevent mistakes.

### 5.2.3 Hidden Files

Please note that there are some configuration files whose name begins with a dot (.). Examples are `.jaleorc`, `.auxchestr` or some project directories (`.IMAGESRC`, `.SOUNDSRC`) whose function will be discussed below. These files are not shown normally in neither the directory views nor using `ls`. To make hidden files visible, activate the Hidden Files option in the View menu of the directory view window, or use the command `ls` with the option `-a` from a shell (`ls -a<enter>`).

## 6. Configuring Jaleo Work Directories

The first important configuration decision you have to make is where you want to place Jaleos work directory.

Jaleo organizes work into projects, created and managed by the Jaleo project manager. A project physically is a directory containing all data pertaining for this particular project, organized in various subdirectories.

You have to configure a work directory for each account you prepare for Jaleo. During installation, a default work directory is created for the original `jaleo` user account, but you can easily reconfigure this as desired.

There always is exactly one active project (the current project) for each Jaleo user. You can have as many projects as you like, but all projects need to be located, at least logically, in a single root directory.

Before we go deeper into that, a short look on a projects structure.

### 6.1 Project Structure

A Jaleo project contains the following subdirectories (if terms used here are not familiar, never mind. For project setup you do not need to know details. See the users manual for more information on the various terms and topics mentioned):

- **CLIPIMAGES.** This directory contains Jaleo clip files. A clipfile is a small text file describing the physical locations for the clips source materials.
- **CLIPSOUND.** The equivalent to Clipimages for sound clips.
- **EDL.** A directory in which EDLs (edit decision lists) are stored.
- **EFFECTS.** Whenever you save an effect parameter setup from the Reel, it will be saved in this directory.
- **ENVIRONMENTS.** In Jaleo, “Environment” is the name for a complete composition work. Whenever you save your work in Jaleo, it will end up in this directory.
- **GALLERY.** Here, Gallery setup files are stored.
- **GROUPS.** When you save a group in the reel, it will be stored in this directory.
- **PAINT.** The work directory for Jaleo RotoPaint.
- **RENDER.** The directory for render scripts. Whenever you select the render function in Jaleo, a render script will be created that can be found in this directory.
- **SETUP.** A private setup directory.
- **SHAPES.** Vector shapes created by RotoPaint belong here.
- **TEXTS.** Texts created by RotoPaint belong here.
- **.IMAGESRC and .SOUNDSRC.** While the clip files stored in the CLIPIMAGES and CLIP-SOUND directories only contain very compact descriptions of where the actual material for a clip can be found, these two directories can contain the image and sound material itself. The “can contain” from the previous sentence is not casual, but very important: Image material can

not only be stored in the directories of the project hierarchy, but also on either raw partitions or connected DDRs. Storage on raw partitions is most of all a performance question - desirable for Jaleo Composite and absolutely necessary for Jaleo Plus, while DDR storage of high resolution images is most of all a question of IO throughput and optimal use of limited disk resources. The material management of Jaleo is described in detail in the introduction section of the users manual. See “Further Disk Space Considerations” on page 10 for more informations.

## 6.2 Project Directory Setup

All Jaleo projects of a Jaleo user are placed in a common WORK directory. The work directory must be placed on a normal file system, but it can be placed on any (locally) mounted disk. Do not use remote disks for the work directory.

Note that a work directory must be configured for each account prepared for Jaleo. By default, when preparing a new account for Jaleo usage using the `makeEnv` program, the new account uses the work directory of the original `jaleo` account as well. This should be changed after setting up the account.

Even if you plan to setup a raw partition as your primary data storage device, you should place your work directory on a drive that has at least one Gigabyte of free storage.

To configure the work directory, first create a directory called WORK (or however you like, although as a convention WORK is recommended) on the disk drive you have designated as your work disk. Then edit the file JALEO\_WORK in the JALEO-ENV directory of the user that is supposed to use this account to include the path to this directory. When the user now logs in to Jaleo, she or he should first start the Project Manager to create a new project.

The file JALEO\_WORK contains a single line of text, being the complete absolute path to your desired work directory.

### Example 1:

Assume the new account prepared for Jaleo is called `salsa`. The account has been created with the user manager using the default location for user directories, i.e. the directory `/usr/people`. This directory is physically located on your system disk and has enough space for a Jaleo work directory.

- Log in as user `salsa`.
- Create the work directory. To do so, open a shell and type:

```
cd <enter>
mkdir WORK<enter>
```

The first command, `cd`, is only given to make sure that you actually are in the home directory of the user `salsa`.

- Now, change to the JALEO-ENV directory to edit the work directory configuration file.  
`cd JALEO-ENV<enter>`
- Open the editor on the file with the work directory description.  
`jot JALEO_WORK`

- A text editor window should appear. In the text file, only one line should be written. Replace that line with the following:

```
/usr/people/salsa/WORK
```

This is actually the complete path describing the location of the work directory that you created previously. Do not forget to press enter after completing the line.

- Save the changes to the file by using the option Save from the editors File menu.

### Example 2:

The same assumption as before, with the single difference that there is not enough space on the disk containing the users home directory. Instead, we assume that there is a second disk drive available, mounted as `/disk2`.

- As a first step, log in as root to give write permissions to the directory `/disk2` to the Jaleo users. As the target directory for the work directory is not inside the users home directory, we can not automatically assume that we have write permissions to the directory (under UNIX, all files are protected by a protection scheme that requires that you have access rights to a file before you are able to read, write or execute it).

After logging in, open a shell and type:

```
chmod ugo+rwX /disk2<enter>
```

This line opens up all permissions to the directory `/disk2` to everyone.

- Now, log out and log in as salsa.
- Create a work directory in `/disk2`. Type
 

```
cd /disk2
mkdir WORK
```
- Change back to your home directory, then to JALEO-ENV to edit the file
 

```
cd
cd JALEO-ENV
jot JALEO_WORK
```
- Enter the path
 

```
/disk2/WORK
```

into the first line of the file. Do not forget to type return after completing the name.

## 7. Configuring Jaleo Operation using the .jaleorc File

Upon start-up, Jaleo reads the file `.jaleorc` in the `JALEO-ENV` directory of the actual user. This file contains a number of option settings that are used to configure Jaleo according to the values in the file.

Not all options need to be present in the file, because there are default values defined for almost each of them. The option file has a very simple syntax:

- Each option is written in one line of the file. You can not specify more than one option in a single line.
- Option names are written in CAPITAL letters. Values like TRUE or FALSE, or ON and OFF, are also written in capital letters. Values that are file names have to be written with the same capitalization as the path itself. UNIX device files typically are all lower case.
- Empty lines are ignored.
- At the beginning of the line comes the option name, then any number of spaces or tabs, and then the value for the option.
- Options can be written in any order.
- Lines or sections of the file can be “hidden” to Jaleo by making them a comment. A comment start with the symbols `/*` and ends with the symbols `*/`. You can not place a comment inside of another comment.

An example for a file is this:

```
/* Configuration for PAL images */
```

```
LOWRESRATIO      4
IMAGESIZEX       720
IMAGESIZEY       576
MAXMEM           32
INTERPOLATION    FALSE
FRAMESSEG        25
NTSCFIELDS       FALSE
NUMLEADZEROS     4
TIMESAVE         300
REALITY          TRUE
RENDERPARTIAL    Raw.dev
CACHEDEVICE      Raw.dev
SCREENRATIO      1.33333333
/* Optional */
/*
CACHEDEVICE      DiskCache.dev
CACHEDEVICE      Raw.dev
```

\*/

Note that the last two configuration lines are *not* read by Jaleo as they are placed inside a comment that starts on the line above and ends on the line below.

A common mistake for configuration files is to forget a terminating line feed at the end of the last line. If you do forget the last line feed, the last line will be ignored. The SGI text editor, `jot`, does automatically warn you if you try to save a file without a new line character at the end. This is a very convenient method to prevent mistakes.

The best way to set options using a `.jaleorc` file is to take the original file, make a copy of it for security and then to change the original.

The `.jaleorc` file is private to each account prepared to use Jaleo. That is, each user can have her or his own private set of settings.

Here an example session to edit the configuration file:

- Log in any account prepared to Jaleo. For this example, let us reuse the account `salsa` used in the examples above.
- Open a shell window and type

```
cd
cd JALEO-ENV
```
- Make a copy of the existing file

```
cp .jaleorc .jaleorc.original
```
- Open an editor on the file

```
jot .jaleorc
```

In the editor window, you should see something quite similar to the example configuration file printed above.

## 7.1 Devices as Configuration Parameters

As some configuration parameters require an value that specifies a “device”, here a quick introduction to the concept. Devices are explained in detail in the chapter on the IO subsystem of the users manual. Device configuration is explained later in this manual in “Configuring Jaleo Devices” on page 62.

Jaleo uses the concept of “devices” to define sources and targets for all kinds of IO operations. A device is a virtual entity that can be used to write to or to read from. Typical IO devices are the various types of supported DDRs. However, Jaleo also handles raw partitions and normal disk filing system directories as devices, i.e. suitable targets for read and write operations. This concept allows for easy extension of Jaleo and facilitates use, as less special cases and special user interfaces are required.

Devices are described by device configuration files. These configuration files are located in the directory `etc/devices` in the `JALEO-ENV` directory.



## 7.2 Valid Options for all Jaleo Systems

The following configuration options are legal inside of the `.jaleorc` file:

### 7.2.1 CACHEDEVICE

The `cachedevice` parameter determines which Jaleo device is used to store cache images. The possible values are the name of a disk or raw partition device file from the `JALEO-ENV/etc/devices` directory. See “Configuring Jaleo Devices” on page 62 for more information on device files and device configuration. Jaleo PLUS systems *always* must be setup for caching using a raw partition.

Possible values for `CACHEDEVICE` are either `DiskCache.dev` or `Raw.dev`. The default, used when the `CACHEDEVICE` entry is not present in the `.jaleorc` file, is the `DiskCache.dev` device. The default file as delivered with the system is set up to use the `/usr/tmp` directory. It is recommended though to redirect this to a dedicated directory of your choice, possibly in the Jaleo `WORK` directory tree.

If you set this option to `Raw.dev`, you *must* configure the raw device configuration file properly to contain a valid raw device path, or your system will not be operational.

Note that even when caching is directed to disk, memory up to the size of `MAXMEM` (see below) can be used for implicit caching of preview material of clips not stored on raw devices. See “Preview Image Storage and Memory Requirements” on page 12 for more information.

### 7.2.2 DATDEVICE

This parameter describes the path name of the UNIX device file to be used to access the systems backup drive (DAT, EXABYTE, DLT). Note that the value of this parameter is *not* a Jaleo device file, but a UNIX device - Jaleo is using the same concept as UNIX in this point.

The backup drive is used by the Jaleo backup and restore application. You must configure the `DATDEVICE` properly if you want to create backups from raw partitions, as you can only do so by using the Jaleo backup software. Files stored on normal file systems can of course also be backed up using the normal UNIX tools.

The default device driver is `/dev/nrtape`. If you have a single tape drive on your system, this device driver file will automatically be created and set up when you boot your machine. If you have multiple backup drives, make sure that either the `/dev/nrtape` device points to the right drive, or that you know the filename of the driver that is used.

Note that `nrtape` stands for “not-rewind tape”. It is VERY important that you, if you use a non-default tape drive configuration, specify a non-rewinding driver for the `DATDEVICE` parameter - a tape driver that rewinds the tape automatically after each operation will not work. Typically, drivers of this type have names like `/dev/mt/tpsXdYnr`, where “X” stands for the SCSI controller number and “Y” stands for the SCSI ID of the drive. You can get informations on connected devices by typing the command `hinv` in any shell window.

In case your system does not have a local tape drive but there is a drive on another system on your network, specify the name of the driver to be used on a remote system. Note that for backup purposes we strictly recommend local tape drives, if possible by any means last generation DLT

(Digital Linear Tape) drives (these are much faster than normal DAT or EXABYTE drives). See the configuration option REMOTEHOST below.

### **7.2.3 EXPIRATIONWARNING**

This variable controls if you see license expiration warnings upon Jaleo startup. You will see a warning dialog box for every temporary license that will expire in a certain number of days. With the EXPIRATIONWARNING option you can configure the number of days before license expiration. If you set this value to 10, you will see warnings whenever you startup a Jaleo applications in the last 10 days before expiration. Setting this value to 0 disables warnings, although it makes sense to set the value on something like 5 days as a minimum.

### **7.2.4 FRAMESSEG**

The number of frames that make up one second of image material. If this parameter is not present, a default of 25 is assumed. The value must be greater or equal to 1. The FRAMESSEG parameter is, among other things, used to define the target framerate for monitor playback and for rendering. Typical values for FRAMESSEG are 24, 25 and 30. In game production, values of 10 or 15 are also common.

### **7.2.5 IMAGESIZEX / IMAGESIZEY**

The horizontal and vertical image size to be used by Jaleo. Note that all input images that are imported should be in this format. If not, the images read in will be clipped or padded to match the input format. Jaleos input/output system does not perform scaling.

The default value for IMAGESIZEX is 720 pixel, corresponding to PAL and NTSC CCIR-601 resolution.

The default value for IMAGESIZEY is 576 pixel, corresponding to PAL CCIR-601 resolution. For NTSC setups, you should set this value to 486.

Note that value setups for NTSC probably also require a change to the NTSCFIELDS option.

### **7.2.6 INTERPOLATION**

This parameter specifies a filtering mode for monitor preview rendering. It does not have any influence on the image quality of a full frame rendering. The default is a bilinear filter, that is applied if the INTERPOLATION value is either not present or set to FALSE. A value of TRUE means bicubic filtering.

### **7.2.7 LOWRESRATIO**

The size of the preview images, relative to the full size images (as specified by IMAGESIZEX and IMAGESIZEY) is specified using this parameter. Legal values are 2, 4, 8 and 16. The value is interpreted as a denominator, thus 2 means half image size, 4 quarter image size and so on.

The default value for Jaleo Composite is 4, for Jaleo PLUS 2. Note that other values can be used for both systems; see “Performance Balancing with LOWRESRATIO” on page 69 for more information. For example, Jaleo PLUS runs very fast with a LOWRESRATIO of 4, saving a lot of storage space, while maintaining acceptable quality for most purposes.

### 7.2.8 MAXMEM

The MAXMEM parameter is the maximum amount of memory that should be used by Jaleo for caching purposes. This includes implicit caching caused by clips that have preview images not stored on raw partitions.

It should be repeated here that disk based caching is highly recommended, if possible by any means using a raw device.

Values given are in Megabytes.

- A normal setting for Jaleo Composite should be one third of the available main memory size. Normally, values higher than 48 are not recommended.
- On systems where you store your preview data more or less exclusively on raw devices, you may lower the value, but it should not be lower than 5.
- In configurations that use a lot of file system storage of preview files, you should raise the value (but to no more than half of main memory size, and to no more than 48).
- For Jaleo PLUS, the value range is about the same (between 5 and 48) - with a sensible default at 32.

### 7.2.9 NTSCFIELDS

Specifies the field order of frames containing independent interlaced fields. As by definition of the CCIR-601 standard PAL and NTSC have different field orders, you must set this parameter to TRUE if you want NTSC style field preference. The default is FALSE, suitable for PAL.

### 7.2.10 NUMLEADZEROS

The number of leading zeroes when creating or reading image sequences from or to a filesystem. Most programs write numbered sequences in a form like

```
test0000.vst test0001.vst test0002.vst etc.
```

In this example, the value for NUMLEADZEROS would be 4. This is also the default value. The minimum value is 0.

Note that the value of NUMLEADZEROS can only be changed on a project-by-project basis. In a single reel you can not mix files with different values for NUMLEADZEROS.

### 7.2.11 PAINT\_TMP\_DIR

The directory for temporary storage of paint. If it is on the same filesystem as your clip directory, always make sure you have enough storage (at least twice the size of the clips you want to process). The default is /usr/tmp, but will typically be changed to a less general file system.

### 7.2.12 TABLET\_MAX\_PRESS

The maximum pressure value a connected tablet can deliver. This value is only used by the Jaleo RotoPaint application. The value depends on the tablet connected and its configuration. The default value is 120.

### **7.2.13 TIMESAVE**

The time interval in seconds between automatic session saves in Jaleo. After the amount of time specified here, Jaleo will automatically create an autosave file of your work. Autosave can be disabled in the Setup menu of the Reel window.

The default value is 300 seconds, or 5 minutes. Shorter values are not recommended.

### **7.2.14 REMOTEHOST**

The REMOTEHOST is used with the Jaleo Backup and Restore applications. In case your system does not have a local tape drive, you can specify here the name of the host to be used for remote backups. Note that you still have to specify the driver to be used on the remote system with the DATDEVICE option.

There is no default parameter; you have to specify a hostname here.

### **7.2.15 RENDERPARTIAL**

Defines the destination device for the group render function of the Jaleo Reel. Although the device specified here can principally be any type of writeable Jaleo device, typically a disk based device should be used here, i.e. either `Raw.dev` or one of the versions of the `Disk.dev` file. Available device files are located in `JALEO-ENV/etc/devices` of the current Jaleo users home directory. See “Configuring Jaleo Devices” on page 62 for more information on device configuration. Note: If you use `Raw.dev` here, do not forget to configure the device file properly for this purpose.

The recommended value for RENDERPARTIAL is `Raw.dev` in case a raw partition is available (always in Jaleo PLUS).

If you have a Cosmo Compress board and at least 128 MByte of RAM, you can also use a `Movie.dev` as target for RENDERPARTIAL. However, the preview images will also be done in JPEG quality, and this is not the optimal in quality. It is better to create movie files with the render option of the IO subsystem.

## **7.3 Valid Options for Jaleo PLUS**

### **7.3.1 PARALLEL**

The number of CPUs available on the machine Jaleo PLUS is running on. On a machine with two processors, this value is 2, on a machine with 4 processors it would be 4 and so forth.

### **7.3.2 REALITY**

This option should be set to `TRUE` if a Reality Engine is present in your Jaleo PLUS system. If so, some zooming operations will be performed in hardware instead of software. The default value of this option is `FALSE`.

## 8. Live Video Configuration Files

The Jaleo reel window and utilities can use available video equipment for output to external video monitors. Specifically, there are three possible applications:

- Any open monitor window can be redirected to video hardware
- Any still frame from a single frame window can be redirected to video hardware
- A Flipbook can be redirected to external video hardware

Currently, this redirection is supported for SGI video hardware (Indy and Galileo video, Sirius video) and, to a limited extent, for DDRs connected to Jaleo. While the SGI video hardware supports all options, i.e. moving images from monitor or flipbook as well as still images from the Single Frame window, DDRs can only support display of still images from the Single Frame window.

The Galileo and Indy video support moving image video playback only for zoomed preview images, not for high res material. In high resolution, only still frames can be displayed.

The Sirius configuration for Jaleo PLUS supports live playback in both full res and previews.

An example configuration file looks like this:

```
DEVICECLASS      LIVEVIDEO
SIZE             10
```

```
/*
```

Configuration Example for DDRs

```
DEVICECLASS      DDR
DEVICETYPE       Abekas.dev
INPOINT          100
```

```
*/
```

The rules for this file are exactly the same as for the `.jaleorc` file: Each line contains one option name at the beginning of the line, followed by a value. Lines included in `/*` and `*/` are treated as comments and ignored. In the example, the DDR configuration is commented out.

### 8.1 Live Video Configuration Options

The configuration file for Live Video is called `ScrToVideo.cfg` and is located in the directory `JALEO-ENV/etc/devices`. It can contain the following options:

#### 8.1.1 DEVICECLASS

The type of device to be used for live video. The available values are `LIVEVIDEO` or `DDR`.

- If the value is `DDR`, only single frame windows can be switched to live video output. A `DDR` requires the presence of the options `DEVICETYPE` and `INPOINT` in the configuration file.

- To use the option Live Video you must have a Indy or Galileo Video or a Sirius Video (Jaleo PLUS only). For the Galileo or Indy Video no further configuration option is necessary. For the Sirius video, a SIZE parameter must be present. Note that the Indy and Galileo video options do not support real time playback for full size video images.

### **8.1.2 SIZE**

This parameter is only required if the DEVICECLASS is set to LIVEVIDEO and the option present is a Sirius board (Jaleo PLUS only). It is the size of the buffer used for the Sirius board and can have values between 4 and 20 frames. As each buffer requires main memory, a value of 4 to 6 usually is a good compromise between performance and memory usage considerations.

### **8.1.3 DEVICETYPE**

This parameter is only required if the DEVICECLASS is set to DDR. The value can be the name of any of the DDR device description files located in JALEO-ENV/etc/devices. In the example above, the description file for Abekas DDRs is used (Abekas.dev). See “Configuring Jaleo Devices” on page 62 for more information on device file configuration.

### **8.1.4 INPOINT**

This parameter is only required if the DEVICECLASS is set to DDR. It specifies at which location of the DDR a single frame is stored that is supposed to be visible on the external video screen. Possible values depend on the size of your DDR; typically, you would use a frame at either the beginning or the end of the DDR.

## 9. Configuring Jaleo for Raw Device Operation

Disk drives used in Raw mode have significant performance advantages compared to normal file system operation. Unless the new `xfs` file system is used (available with a special edition of IRIX 5.3), normal file systems also pose a limit on the maximal disk space usable as a single logical unit (see “Raw Devices Versus `xfs`” on page 14 for more information).

For an informal description of raw devices, please see the end of the introductory chapter of the users manual. There is also some information in the chapter on the IO subsystem.

Raw partitions offer a convenient way to access disks or disk arrays in the most efficient and elegant way. This is why raw partition storage is the optimal option for maintaining image data for Jaleo Composite, and a mandatory storage location to achieve realtime full resolution uncompressed video IO with Jaleo PLUS. **For realtime performance with Jaleo PLUS, the raw partitions need to be set up as a logical volume using at least 4 controllers and 8 disks.**

To use a raw device, you have to do the following steps:

- Prepare one or multiple disks partitions
- If you want to use multiple disks, create a logical volume from the disks involved
- Configure the raw device file in `JALEO-ENV/etc/devices`

While the first two steps are discussed in this chapter, please see “Configuring Jaleo Devices” on page 62 for information on device file configuration.

**Please remember that raw device configuration, especially for Jaleo PLUS, is far from trivial and thus should not be attempted by unexperienced users. If you are not very fluent in UNIX operation, please do not attempt to setup a raw device but ask your dealer or distributor for a properly configured system. An incorrectly setup raw device can jeopardize Jaleo operation completely; Jaleo PLUS systems with improperly configured disk arrays will not produce real time performance.**

### 9.1 Differences in Jaleo Operation Imposed by Raw Devices

Raw devices impose a few differences in operation on the Jaleo system. The reason for these changes are all rooted in the fact that the content of a raw device is not accessible to the normal UNIX system tools.

- A raw device must be configured as described below.
- Jaleo must be configured to use raw devices. The most important configuration options are `CACHEDEVICE` and `RENDERPARTIAL` in the `.jaleorc` file (see “Configuring Jaleo Operation using the `.jaleorc` File” on page 45).
- When creating clips with the non-realtime IO options, make sure you select the raw storage option for the clips to prevent implicit caching with all its negative impacts on performance due to excessive memory usage (see “Preview Image Storage and Memory Requirements” on page 12 for more information).

- The real time capture application for the SIRIUS video (Jaleo PLUS only) always captures to a raw device disk array of appropriate performance, i.e. able to write securely 42 MB per second in PAL mode.
- To list the content of a Jaleo raw device, a special tool called `xlsRaw` must be used. See the Jaleo release notes for more information.
- To delete clips stored on raw devices, you must use the `dustbin` application. `dustbin` can also be used to comfortably delete Jaleo files from other storage devices. See the Jaleo release notes for more information.

## 9.2 Understanding Device Path Names

Before we proceed with the disk configuration, a few comments on device control under UNIX. Each disk drive connected to a UNIX system is controlled by a so called device driver. A device driver is located in the file system just like a normal file. All UNIX device drivers are located in the `/dev` directory.

There are two types of drivers for each disk. One driver is used to access a disk after it has a file system placed on it. These drivers are located in the

`/dev/dsk`

directory for SCSI disks. For SCSI disks accessed as raw partitions, these drivers are located in

`/dev/rdisk`

Disk drives themselves are named like this:

- First, there is an abbreviation for the controller type. For SCSI disks, this abbreviation is `dksc`.
- Then, there is a number to identify the SCSI controller. Many SGI systems can have more than one SCSI controller. For the normal internal system disk, the controller number is 0.
- Then, there is the single letter `d` followed by a number for the drive id. SCSI drive ids can be numbers from 1 to 7.
- Finally, there comes the letter `s` followed by the partition number. For raw device setup, typically the device number 7 is used. By convention, partition 7 is the partition that uses all off the disk, with the exception of a very small volume header partition that always is required.

The full name for a SCSI disk to be accessed as a raw device, connected to SCSI controller 2 as device number 4 is thus:

`/dev/rdsk/dks2d4s7`

This is actually the name of the driver file that is used to access the disk, but as this is the most immediate access for a disk under UNIX, we will just use it synonymous with the disk itself.

Note: For multiple drive logical volume configurations, typically the driver files in `/dev/dsk` are used (instead of the drivers in `/dev/rdsk`). See “Creating Raw Devices from Multiple Disks” on page 58 for more information.



## 9.3 Possible Raw Device Configurations

Raw devices can be either single disk drives or striped arrays of drives (the latter is a must for Jaleo PLUS). Striping refers to the mode of use in that information is distributed over multiple disks so that as many drives as possible can be used at the same time to access a piece of information, in effect speeding up disk access massively. Of course, if multiple disk striping is done with disks connected to more than one controller, performance can be increased even more, as a single controller can only deliver a limited amount of data. For realtime digital video without compression, in PAL mode roughly 42 MByte per second need to be read or written. This requires at least four SCSI controllers with and 8 disk drives, with 2 drives connected to each controller.

It is strictly recommended always to use a complete disk for raw device storage, i.e. partition 7 of the disk. For performance reasons, this is mandatory in striped disk configurations. In particular, never try to create two raw partitions on the same drive.

### 9.3.1 Single Drive Raw Device Configuration

**The following should only be attempted by experienced UNIX users. Please ask your dealer or distributor for a properly configured system.**

To configure a single drive raw device, not much work is required. However, if you are not familiar with configuring a computer be *very* careful here. If you accidentally apply disk configuration changes to your system disk, you will have to reconfigure your computer and all data on you system disk will be lost. We recommend to ask your dealer or distributor for a proper configuration of your system.

The disk needs to be connected physically to the computer. Switch off the computer and all peripherals before connecting the drive. Use a SCSI ID that is not yet used by other peripherals and select the SCSI controller that is least used by other equipment. On an Indy, you do not have a lot of choices, unless you buy an additional SCSI controller card, on an Indigo 2 you should use the external controller.

After connecting the drive, switch on the computer and watch the boot process. Click the system maintenance button when the system displays its boot message (on older systems, you may have to press the Escape key instead when prompted) shortly after power on.

In the monitor menu that appears now, select System Monitor. In the shell-like window that appears, type:

```
hinv
```

Make sure that both the system disk and the newly connected disk are visible in the list of connected drives. If not, you should power down the system and consult your dealer or distributor for help.

Press the Exit button on the lower right corner of the system monitor (or type exit on older systems) to go back to the Monitor menu. Select Start System. Wait until the log in screen appears and log in as root.

Open a shell window. Type

```
fx -x
```

to invoke Silicon Graphics disk maintenance tool. `fx` will ask you a number of questions before presenting its initial menu.

- The first question is the type of controller your disk is connected to. This will almost always be SCSI, so the default value (`dksc`) is Ok. Just press return.
- The second question is the controller number. On an Indy, this always is 0, on an Indigo 2 it may be either 0 (for the internal controller) or 1 for the external controller. On an ONYX system, there are many possibilities. If you are not sure where disks are connected, open another shell window and type `hinv`. This will print out a hardware inventory list that among other things gives you the SCSI controllers and connected disk drives.
- The third question is the drive ID. Be careful not to accidentally specify the system disk or any other mounted data drive with a filing system. If you do so, `fx` will print out a warning like this:

“Warning: This disk appears to have a mounted filesystem...”

In this case, exit `fx` immediately by typing `exit` and start again.

After selecting a new fresh disk drive, `fx` will first test the controller and then complain about the fact that disk does not yet have a valid SGI volume label. In this case, `fx` is perfectly right, as a fresh disk drive can not have one. `fx` will also inform you of the fact that it created a default label before presenting you with its main menu.

- From the `fx` main menu choose the menu option `repartition`. In `fx` menus, the characters you have to type are always included in square brackets. To go to the `[r]epartition` menu, it is thus enough to type `r`.
- In the partitioning menu, you will see a listing of the current partitioning layout of the drive. A description of partitioning schemes is not in the scope of this manual; it should be noted though that there always are two partitions, labelled 8 and 10, that constitute a volume header area of 2 Megabyte that is used for disk administration, and the full disk area that can be used to create 1:1 copies of a disk to another physically identical drive. Partition 10 thus overlaps all the other partitions that may be present. It is never used in normal operation. Note that in general it is possible to create overlapping partitions; you must *never* use two overlapping partitions at the same time, as this will certainly ruin your data.
- Choose the option  
`[o]ptiondrive`

from the `fx` menu. If you confirm upon the request given to you, there is no way back - if there was any data on the drive before, it will be irrevocably lost. The `optiondrive` option creates a disk with a single usable partition that covers the whole disk. By convention, on SGI systems this partition has the partition number 7.

The system will display the new partition layout. For an `optiondrive` configuration, you should have three lines displayed (see “Creating Partitions” on page 9 for some more information on partitions):

- One for partition 7,
- one for partition 8 and
- one for partition 10.

- Please write down the total blocksize of partition 7, as this size will later be needed to configure the device description file. In the table printed by `fx` on the new partition layout there is a column `blocks`. The blocksize is given in the form `<starting block> + <block size>`. The number you need to use is the one on the right side of the `+` sign. For a diskdrive of 2 GB, a possible value would be

`3990+4246690`

The size you need is the `4246690`.

If you forget the number, you can of course re-run `fx` to simply list the current partitioning. Note that the size is only needed for single disk raw device configurations.

- Go back to the `fx` main menu. You can do so by typing to points followed by enter.

`..<enter>`

- Go to the label menu

`[l]abel`

- Use the option

`[sy]nc`

to write out the changes to the disk drive.

- Go back to the main menu and exit `fx`. In case you want to proceed with another drive, you can type two dots (`..`) to have `fx` asks you to choose another drive.

The drive is now prepared to be used as a raw device. You can use the device name of the drive immediately in the Jaleo device configuration file that sets up the raw device for Jaleo. Before you can use the raw device in Jaleo, you have to configure the Jaleo device description file. Please proceed to “Configuring Jaleo Devices” on page 62 for more information on device file configuration.

After configuring the `Raw.dev` Jaleo device configuration file, do not forget to modify the entries `CACHEDEVICE` and `RENDERPARTIAL` in the `.jaleorc` file for Jaleo to use the raw device.

### 9.3.2 Using a Partition on a Non-Dedicated Disk as Raw Device

**The following should only be attempted by experienced UNIX users. Please ask your dealer or distributor for a properly configured system.**

Any partition on a disk drive connected to an SGI workstation that is currently not in use by a filing system can be used as a raw device without any further preparation. You can specify the device driver name for the unused partition in the device configuration file (see “Configuring Jaleo Devices” on page 62)

As an example, let us assume you have a disk configured with a root partition (By convention, partition number 0), swap space (partition number 1) and a user partition (partition number 6) that you can spare. In this case, the device name would be:

`/dev/rdisk/dks0d1s6`

provided the disk is connected to controller 0 as drive id 1. We would like to repeat that this is not a recommended configuration. It is always recommended to use a dedicated disk as raw partition.

You can use the device name of the drive immediately in the Jaleo device configuration file that sets up the raw device for Jaleo. Before you can use the raw device in Jaleo, you have to configure the Jaleo device description file. Please proceed to “Configuring Jaleo Devices” on page 62 for more information on device file configuration.

### 9.3.3 Reusing a Disk that has a File System as Raw Partition

**The following should only be attempted by experienced UNIX users. Please ask your dealer or distributor for a properly configured system.**

A partition with a file system can simply be reused. If you are sure that you do not need any of the data on the disk, you can remove all `fstab` entries referring to the disk drive. Please see the manual pages for more information on `fstab`. To give a quick summary, this is a file on your system that contains information about all file systems to be mounted automatically. You must *never* attempt to access a disk that is currently mounted via its raw mode driver, i.e. via the Jaleo raw device driver.

After the partition is unmounted and all references are removed from the mounting tables, you can use the disk path using the raw device driver file in `/dev/rdisk` in the Jaleo raw device configuration file (`Raw.dev`) as usual.

It is recommended to use a dedicated disk as a raw partition. If possible by any means, do not share a raw partition drive with other partitions.

You can use the device name of the drive immediately in the Jaleo device configuration file that sets up the raw device for Jaleo. Before you can use the raw device in Jaleo, you have to configure the Jaleo device description file. Please proceed to “Configuring Jaleo Devices” on page 62 for more information on device file configuration.

### 9.3.4 Creating Raw Devices from Multiple Disks

**The following should only be attempted by experienced UNIX users. Please ask your dealer or distributor for a properly configured system.**

Multiple disk drives can be used together to form a single *logical volume*, resembling a huge disk drive with the summed capacity of all participating drives. All drives participating in the logical volume need to have exactly the same capacity. For best performance, it should be the same drive type.

- As a first step, use the `fx` tool as described above for a single disk (see “Single Drive Raw Device Configuration” on page 55) on all the disk drives to be used in the array to prepare the partitioning. Use the `optiondrive` partitioning scheme for all drives.

### Creating a Logical Volume

To create a logical volume, a description file must be created to inform UNIX which drive partitions are used for the logical volume. Then, the logical volume must be initialized using a special SGI tool. As we want to use the logical volume as a raw device, there is no need to create a file system on the logical volume.

The description file for logical volumes is called `lvtab` and it must reside in the `/etc` directory. On a system freshly configured, typically this file does not exist.

To create or edit the file, log in as root (you still should be root from all the steps above)

- Open a shell window and type

```
cd /etc
```

- Open an editor on `lvtab`

```
jot lvtab
```

The editor window that appears will typically be empty, as the file in most cases does not exist. In `lvtab`, the specifications of all the logical volumes attached to the machine are located. Logical volumes are usually named `lv0`, `lv1`, `lv2` etc. A full description of the format of `lvtab` entries can be found in the associated man page (type `man lvtab` in a shell window). See also `lvinit` and `mklv`.

All disk partitions used in a logical volume *must be identical* in block size.

### An `lvtab` Entry for Jaleo Composite

An example `lvtab` entry for a simple logical volume of two disk drives for an Indigo2 system:

```
lv0:JaleoYeah:stripes=2:step=1620:\  
devs=/dev/dsk/dks1d2s7, /dev/dsk/dks1d3s7
```

The entries have the following meaning:

- `lv0` is the device identifier for the logical volume. If you have multiple logical volumes in your `lvtab` file, you should give them consecutive ids, i.e. `lv0`, `lv1` etc.
- The next entry is a more readable device name
- The `stripes` parameter determines how the data should be distributed over the disks. For our purposes, the `stripes` parameter typically has a value that is identical to the number of disk drives.
- The `step` parameter is very important for performance. Although its value is not of uttermost importance for Jaleo Composite systems, it is **crucial to real time performance with Jaleo PLUS**. See below for a formula to calculate the step size.
- Note that if you want to split an `lvtab` entry into multiple lines (you can do so after any of the `:` characters), you have to insert a backslash character (`\`) as the last character of the line.
- The `devs` parameter is a list of the disk drivers to access the disk partitions. Note that here the drivers in the `/dev/dsk` directory are used.

After editing the file (do not forget to press enter after the last line entered) save the content by using the Save option from `jot's` File menu.

### An `lvtab` Entry for Jaleo Plus

**For Jaleo Plus, proper setup of the logical volume is of uttermost importance for real time performance. A Jaleo PLUS system with an improperly configured disk array will not produce real time playback or capture. Setup is not trivial. If you are not a nexperienced UNIX user, please ask your dealer or distributor for a properly configured system.**

You have to use logical volumes that consist not only of more disks, but the disks also need to be connected to multiple controllers. For the current generation of disk drives, typically at least 4 controllers and 8 disk drives are required. The `lvtab` entry is very similar to the example above. The critical configuration parameters for this setup are the `step` parameter and the order of drives in the `devs` list.

An example of an `lvtab` for an 8 disk array configured for PAL:

```
lv0:Jaleo Team:stripes=8:step=405:\
devs=/dev/dsk/dks0d2s7, /dev/dsk/dks1d2s7,\
/dev/dsk/dks6d2s7, /dev/dsk/dks7d2s7,\
/dev/dsk/dks0d3s7, /dev/dsk/dks1d3s7,\
/dev/dsk/dks6d3s7, /dev/dsk/dks7d3s7
```

**The order of partitions is crucial.** Cycle first through the controllers, then through the attached drives. That is, first refer to the same drive number (for example 2) on all the different controllers, then repeat the same with the next drive number etc.

Remember that you can find out about the controllers and disks present in your system by using the `hinv` command.

- After entering the entry, do not forget to store the file to disk by using the Save option from the File menu.

If you have a different number of disks, you have to change the number of `stripes` given above.

For real time performance, the `step` value is very important. It depends on the size of the images you want to store and the number of drives available. You can calculate the `step` size using the following formula:

$$((\text{image } x * \text{image } y * 4) / 512) / \text{noOfDisks}$$

Example values for CCIR601 PAL images are 405 and for NTSC 342.

## Setting Up Logical Volumes With RAID Arrays

Note that calculating these logical volume parameter values for RAID arrays may work totally different depending on the specifications of the manufacturer. The formula given above applies to striped arrays of normal disk drives only. At the present date, trying to run Jaleo PLUS with RAID arrays is *entirely at your own risk*. We do currently not recommend the use of RAID arrays.

## Preparing the Logical Volume for Use

- After you have created the `lvtab` entry, the logical volume must be initialized with `mklv`. Any syntax errors in `lvtab` will be printed out by `mklv`. Type:

```
mklv lv0
```

Of course, if you have used a different logical volume identifier (e.g. `lv1` etc), you have to change the commandline appropriately.

If `mklv` gives any errors, please see the man page on `mklv` and `lvtab` for more information. `mklv` does refuse to create logical volumes from partitions that are already in use either as a file system or as part of another logical volume.

### Using the Logical Volume

After creating a logical volume with `mklv`, the volume can be addressed using the device driver path

```
/dev/rdisk/lv0
```

with the `lv0` identifier possibly being replaced by the identifier you chose for your logical volume.

The logical volume is now prepared to be used as a raw device. You can use the device name of the drive immediately in the Jaleo device configuration file that sets up the raw device for Jaleo. Before you can use the raw device in Jaleo, you have to configure the Jaleo device description file. Please proceed to “Configuring Jaleo Devices” on page 62 for more information on device file configuration.

After configuring the `Raw.dev` Jaleo device configuration file, do not forget to modify the entries `CACHEDEVICE` and `RENDERPARTIAL` in the `.jaleorc` file for Jaleo to use the raw device.

### 9.3.5 Multiple Controllers on Indigo2 Systems

Some third party vendors offer special add-ons for the Indigo 2, allowing to access the internal SCSI controller from external drives, making striped arrays on multiple controllers possible on an Indigo 2. If you have such a kit, you can proceed creating a logical volume as described in the section above.

## 10. Configuring Jaleo Devices

Jaleo uses the concept of “devices” to define sources and targets for all kinds of IO operations. A device is a virtual entity that can be used to write to or to read from. Typical IO devices are the various types of supported DDRs. However, Jaleo also handles raw partitions and normal disk filing system directories as devices, i.e. suitable targets for read and write operations. This concept allows for easy extension of Jaleo and facilitates use, as less special cases and special user interfaces are required.

Devices are described by device configuration files. These configuration files are located in the directory `etc/devices` in the `JALEO-ENV` directory. Before you can use a device, the appropriate device file must be configured to meet your particular system setup.

Most device files are exclusively used inside of the Jaleo IO subsystem. Some device types, like the Raw or DiskCache device are used by Jaleo in a more elementary fashion. You already encountered some of these device files during the configuration process. If you have configured Jaleo for Raw device or Disk device caching in the `.jaleorc` file, you **MUST** configure the appropriate device file or Jaleo will not operate.

Some configuration files contain preset options, i.e. options that can be overridden in the IO subsystem user interface as well as mandatory options that have to be specified in the configuration file. An example are for example the DDR devices that permit you to specify connection options in the device file, although these options can be overridden in the interface. Other options, for example to specify the imagesize supported by the DDR, can not be changed in the interface.

If you want to have multiple presets of a given device, you can copy any of the device files and give it another name. Device files principally can have any name that ends in the extension `.dev`. However, drag and drop is only supported properly by the device files if the first characters remain the same. For example, you can create a copy of the default file `Abekas.dev` with the name `AbekasA65.dev` and one with the name `A65.dev`. Both files will be operational inside of the IO interface, but only the file named `AbekasA65.dev` will support drag&drop.

Note that the original configuration file names should not be renamed. You can make copies, but devices with the original names always should be maintained.

### Exception for Raw.dev

The raw device configuration file, `Raw.dev`, is a unique exception from the rule. As only one raw device can be in use by Jaleo and there are no preset-able parameters for the file, you can not use another copy of the file. *Never* try to use multiple raw device configuration files as this will corrupt your raw device content.

### 10.1 Device Configuration File Syntax

The device configuration files use the same syntax as the other configuration files discussed above. To repeat:

- Each option is written in one line of the file. You can not specify more than one option in a single line.



- Option names are written in CAPITAL letters. Values like TRUE or FALSE, or ON and OFF, are also written in capital letters. Values that are file names have to be written with the same capitalization as the path itself. UNIX device files typically are all lower case.
- Empty lines are ignored.
- At the beginning of the line comes the option name, then any number of spaces or tabs, and then the value for the option.
- Options can be written in any order.
- Lines or sections of the file can be “hidden” to Jaleo by making them a comment. A comment start with the symbols `/*` and ends with the symbols `*/`. You can not place a comment inside of another comment.

This is an example file:

```
DEVICETYPE      ABEKAS
MESSAGES        ON
FORMAT          NTSC
USER            guest
HOSTNAME        ABEKAS
IMAGESIZEX      720
IMAGESIZEY      486

CONNECTION      SCSI
PORT            /dev/scsi/sc1d6l0

/*
CONNECTION      ETHERNET
PORT            192.1.1.30
*/
```

This is an example file for an Abekas DDR configured for SCSI connection. At the bottom, an ethernet connection setup is commented out.

## 10.2 Common Device File Options

All Jaleo device files *must* at least contain a devicetype field.

### 10.2.1 DEVICETYPE

The type of device. Possible values are currently (see the release notes for possible deviations):

- ABEKAS
- ACCOM
- CLIP

- DISK
- HP
- MOVIE
- NULL
- QUICK
- RAW
- RENDER
- VTRACCOM

### 10.2.2 MESSAGE

The message option determines the amount of messages that is generated by the device if used in the IO subsystem. A value of `FALSE` generates only minimal messages, while a value of `TRUE` is more verbose.

## 10.3 Basic Configuration Devices

### 10.3.1 DiskCache.dev

The DiskCache device is a special case of the Disk device. Actually, it is just a standard disk device file, setup with parameters suitable for caching. A disk device configuration file with the name `DiskCache.dev` always must be present in the `etc/devices` directory. A disk device used as a value for the `CACHEDEVICE` option of the `.jaleorc` file must have the following four options customized):

#### DEVICETYPE

The type for the DiskCache device must be `DISK`.

#### PATH

A file system path where cache images should be stored. The default file delivered with Jaleo specifies `/usr/tmp`. You should set this parameter to a directory path to a directory you create on a disk file system with sufficient disk space (>100 MB). Again, it is better to use raw partition caching if possible by any means.

#### INPUTFORMAT

For a disk device to be used as Cache, the value should be `VST_2`.

#### OUTPUTFORMAT

For a disk device to be used as Cache, the value should be `VST_2`.

### 10.3.2 Raw.dev

The raw device configuration file, by default called `Raw.dev`, should always be the only device configuration file that is of type RAW. Never use two of these files - you will probably corrupt your data. You *must* configure this file properly if you have set any option in the `.jaleorc` file to use raw devices (CACHEDEVICE, RENDERPARTIAL).

#### DEVICETYPE

For a raw device, the device type is RAW.

#### PATH

Use the device path to the configured raw partition or logical volume. You should have set up the raw partition or logical volume as described above (see “Configuring Jaleo for Raw Device Operation” on page 53). For use with logical volumes, a typical value would be `/dev/rdisk/lv0`, an example for a single disk would be something like `/dev/rdisk/dks0d3s7`.

#### SIZE

The size parameter is of uttermost importance.

- For a single disk raw partition, you must here use the total blocksize of you raw partition as it was printed out by `fx` during the configuration. If you have not written down the block size of your disk partition, use the `fx` command again to find out (use the `label/show/partitions` option in `fx` to display the partition table).
- For a logical volume, use the command `lvinfo` to get the size of the device. Type  
`lvinfo lv0`  
if you created your logical volume with a volume id of `lv0`, or use the id you defined in the `lvtab` entry instead.

For more information, see “Configuring Jaleo for Raw Device Operation” on page 53.

## 10.4 Clip Device

The only parameter, except messages, for clip devices is the DEVICETYPE. Therefore, there is no room for much customization.

#### DEVICETYPE

The device type for clip devices is CLIP.

## 10.5 DDR Devices

All DDR devices basically support the same set of options. We thus will give a generic explanations, mentioning differences between the various options on the way.

## **DEVICETYPE**

The devicetype for DDRs can be ABEKAS, ACCOM, VTRACCOM, HP or QUICK. “VTRACCOM” refers to an Accom using VTR control, “QUICK” refers to the Sierra Design Labs (SDL) Quick-frame recorder.

## **FORMAT**

The format field can have a value of either NTSC or PAL.

## **USER**

The user field contains the name of a user that has permissions to operate on the DDR (this must be configured on the DDR). It is only important to specify a valid user for the ABEKAS device type.

## **HOSTNAME**

The name of the host as defined in the /etc/hosts network configuration file. This entry is only required if the DDR is connected via Ethernet.

Exception for Abekas DDRs: If an Abekas connected via SCSI is to be used to output Single Frames to external video it must be connected via Ethernet as well. However, in this case the CONNECTION (see below) would be specified as SCSI, as the Ethernet address can be found via the HOSTNAME.

## **IMAGESIZEX / IMAGESIZEY**

The size in pixels of transferred images in horizontal and vertical direction. It is very important to specify the values correctly according to the DDR type.

## **SIZE**

The storage size in frames. For the ACCOM DDRs, the correct SIZE setting is mandatory. For the other DDRs, it is only used for error checking. Due to a problem in the current Accom software, for the Accom WSD XL models the same size must be specified than for the old Accom WSD. All frames will be usable, even with the value at the lower setting for the old WSD.

## **CONNECTION**

The type of connection to be used. This can be either SCSI or ETHERNET. The Accom WSD supports a third connection type, GIO.

## **PORT**

The specification for the connection specified above.

- In case the CONNECTION field has the value of SCSI, the value is a SCSI device driver:

/dev/scsi/sc1d6l0

In this example, this is a SCSI device connected to controller 1 with ID 6.

- In case the CONNECTION field has the value of ETHERNET, the value is an Internet address in “dot notation”, for example:  
192.1.1.17

## 10.6 Disk Device

With the disk device, it makes most sense to have multiple copies configured with different PATH, INPUTFORMAT or OUTPUTFORMAT options. This allows to quickly access the respective directories and/or formats, without a lot of input activity in the IO program.

### DEVICETYPE

The device type is DISK.

### PATH

The default path shown in the IO interface for this device type. If the path is not an absolute path (i.e if it does not begin with a slash character “/”), then it is interpreted to be relative to the currently active project.

### INPUTFORMAT / OUTPUTFORMAT

The image format shown as default if this device is selected for input/output in the IO subsystem.

The options currently supported are:

- ALS. Alias files
- IM. Chyron Liberty files
- PIC. Softimage files
- RGB. Silicon Graphics files
- RLA. Wavefront files
- TIFF. Tiff files
- VST\_2 or VST. Uncompressed Targa/Vista images
- VST\_10. Compressed Targa images
- YUV. 16 Bit YUV files as for example produced by DDRs

### IMAGESIZEX / IMAGESIZEY

For formats that do not include size information (for example YUV) this information gives the size to be expected in X and Y.

## 10.7 Movie Device

### DEVICETYPE

The device type is MOVIE.

## **PATH**

The default path shown in the IO interface for this device type. If the path is not an absolute path (i.e if it does not begin with a slash character “/”), than it is interpreted to be relative to the currently active project.

## **OUTPUTFORMAT**

The compression format shown as default if this device is selected for output in the IO subsystem. For input, the format is recognized automatically

The options currently supported are:

- JPEG
- MVC2
- UNCOMPRESSED

## **10.8 Render Device**

The only parameter, except messages, for render devices is the DEVICETYPE. Therefore, there is no room for much customization.

## **DEVICETYPE**

The device type for render devices is RENDER.

## 11. Balancing Performance in Operation

Jaleo operation and performance can be balanced easily to match the requirements of a particular job. This section will repeat some of the configuration options that you can use to optimize your production work:

### 11.1 Choice of Material

Jaleo offers the capability to mix various types of material in a single production. You can mix freely uncompressed image material of full CCIR-601 quality with JPEG compressed material captured using the Cosmo Compress hardware option. Even mixed operation with different compression ratios from one clip to the other is possible. Also, you can use software compressed material.

Please note that although you can use different *types* of material, you should make sure that all material for a single project has the same *resolution*, as configured in the `.jaleorc` file (IMAGESIZEX/IMAGESIZEY).

Selecting the right type of material can raise your production efficiency and lower your costs at the same time. For example, long background shots often do not suffer at all from JPEG compression, while bluebox shots can not be compressed at all whenever proper key quality is to be maintained. Jaleo does not force you to hold all material on the quality level required by the most demanding operation. You can have each shot in the material type required by the operation, or in the type that is most convenient for the job.

### 11.2 Material Storage

Depending on your system setup, you can store clip material where it is most appropriate. You can leave it on an external DDR to save the upload time and to extend the available storage total. You can store it on a raw device for the most rapid access. You can store data on a normal disk file system to make use of the convenient access to such material by all available tools.

You should be aware, though, that clips that store their preview images on file systems may have higher demands on system memory resources. See “Preview Image Storage and Memory Requirements” on page 12 for more information.

### 11.3 Performance Balancing with LOWRESRATIO

On a project-by-project basis you can balance speed of interactive operation versus preview quality and storage demands with the LOWRESRATIO parameter. For Jaleo PLUS, the default value is 2, i.e. the preview images are half size images, for Jaleo Composite by default quarter resolution is used. While inside a single project you must use the same LOWRESRATIO for all clips, you can easily switch between projects.

For a project with large volumes of data, or that requires a lot of quick assembly and editing, you can easily switch LOWRESRATIO to a value of 8 (or 4, for Jaleo PLUS). This gives a significant increase in performance, while lowering storage demands. If you require higher quality during operation, you can switch to half res previews for Jaleo Composite or even work directly with the full res material in Jaleo PLUS. Although this will be slow, the quality benefits will sometimes

make it worthwhile. Note that using the full res option of the monitor you can switch to full res operation temporarily while working in a project.



## 12. Clip File Format

The following is a quick description of the Jaleo clip file format of Jaleo version 2.x. Jaleo 2.x can still read the old format of Jaleo 1.x, but always writes clip files in the new format.

Clip files are simple text files that can be edited with any text editor.

The following is an example clipfile:

```
C(
  f(0.000000) l(25.000000)
  p(
    n(ltrio)
    X(2305) Y(160) W(25) F(0)
  )
  d(
    I(
      s(0) e(24) i(0)
      L( d(Disk.dev) p(MPROJECT/.IMAGESRC/lTRIO/) n(ltrio%04d.md))
      H( d(Disk.dev) p(MPROJECT/.IMAGESRC/lTRIO/) n(ltrio%04d.vst))
    )
  )
)
```

### 12.1 Basic Format

A clip file is made up from nested labelled sections. A label is a single character, and a section is opened by an opening round bracket and closed by a closing round bracket. A section can, dependent on its label, either contain other labelled sections, or direct values. As a section always is enclosed in brackets, a value is expressed as a label followed by a value in brackets, for example `f(12.000000)`.

Values can be strings, floating point or integer numbers.

#### Toplevel Section: Label “C”

A clip file always consist of a single toplevel section. The label for clip toplevel sections is C for “Clip”. The bracket pair following the C section must enclose all other content of the clipfile.

### 12.2 Clipfile Sections

The clip file sections are best explained following the example above.

#### 12.2.1 Instance Data

The first sections inside of the toplevel section contain clip instance data. That is, they do describe which frames of the clip source material are actually used by the clip, as well as some other properties that are independent of the actual source data. Clip instances thus can denote trimmed clips.

### **First Instance Frame: Label “f”**

After the label `f ( )` in the example above follows the number of the first frame of the source material that is used by the particular clip instance. Note that the first instance frame can have a number larger than the last instance frame (see below), denoting a flipped clip.

### **Last Instance Frame: Label “l”**

After the label `l ( )` in the example above follows the number of the frame following the last frame used of the source material of the particular clip instance. That is, it is the first frame that is not used any more. If the last instance frame number is smaller than the first instance frame number, the clip is flipped.

### **12.2.2 Property Section: Label “p”**

The property section, enclosed in the label `p ( )`, contains a number of attributes for the clip instance. These include name, reel position, width and interlace flag.

#### **Name: Label “n”**

The name specified after the `n ( )` label is the instance name that appears in the reel icon when the clip is loaded into a reel window.

#### **Reel Position: Label “X” and “Y”**

The labels `X ( )` and `Y ( )` are used to specify position in an arrangement. Unless the clip is part of an environment file, these values can safely be ignored.

#### **Absolute Instance Length: Label “W”**

The `W ( )` label specifies the length of the clip instance in frames.

#### **Active Frame: Label “F”**

The `F ( )` label specifies the frame of the clip the clip preview cursor is positioned at.

### **12.2.3 Source Data Section: Label “d”**

The `d ( )` section encloses all information referencing the source material of the clip. It contains only a single section, either describing an image clip (label `I`) or a sound clip (Label `S`).

#### **First Source Frame: Label “s”**

The `s ( )` section contains the number of the first frame present in the source material.

#### **Last Source Frame: Label “e”**

The `e ( )` section contains the number of the last frame present in the source material.

### **Interlaced Flag: Label “i”**

The value inside the `i ( )` label is 0 if the clip is not interlaced, or 1 if it contains interlaced frames.

### **12.2.4 Source Location Sections Label “L” and “H”**

The source location sections have the same content format. The `L ( )` section describes the location and format of the preview (low res) material, and the `H ( )` section describes location and format of the full (high) res material.

Each of the source location sections contains three labels:

- A device label, `d ( )`, containing the name of the storage device
- A path label, `p ( )`, containing the path name to the image material
- A name label, `n ( )`, containing the name and numbering scheme of the image material.

#### **Source Device Name: Label “d”**

The device name describes a device on that the data is stored. Possible values are a disk device, a raw device or a DDR device. The interpretation of the following `p ( )` and `n ( )` labels depend on the storage device.

#### **Source Path: Label “p”**

The `p ( )` label contains a path describing the location of the images on the given device.

- If the device is a disk device, the path is relative to the Jaleo WORK directory.
- If the device is a raw device, the path is a pseudo filename inside of the raw partition.
- If the device is a DDR device, the path is a number specifying the offset of the sequence, i.e. the first frame number used by the sequence.

#### **Source Name: Label “n”**

The `n ( )` label content interpretation depends on the device type.

- For a disk device, the name is the base file name plus a numbering scheme given in “printf” notation plus the file extension. The numbering scheme begins with a percent sign, contains as an option the number of leading zeroes and ends with the letter “d”.
- For a raw device, the name is only a numbering scheme as described above.
- For a DDR, only a numbering scheme is given, of the format `%05d`.