

1. Introduction

This portion of the manual is an addendum to the users manual. It contains supplementary information on a number of tool programs from four categories:

1. Real-time IO for Sirius Video (Jaleo PLUS) and Galileo Video/Cosmo Compress (Jaleo Composite)
2. Jaleo Backup and Restore Applications for quick and easy backup and restore of Jaleo data. On Jaleo PLUS systems, the Jaleo Backup and Restore applications can create backups of material in video format on tape recorders, thus greatly increasing backup performance compared to traditional backup devices.
3. File Management Tools contain applications to retrieve information and status of raw partitions and a deletion tool for Jaleo clips including the referenced image data. This deletion tool gives you a reference list of other files using the same material, greatly easing the task of deciding if a clip with its associated material indeed is not needed any more.
4. License Management. Jaleo 2.1 is using the Élan License Management system that provides very comfortable and easy licensing and. A number of applications are provided with Jaleo to help you manage your network licenses.

2. Realtime IO for Sirius and Cosmo

The realtime IO applications for Sirius Video and Cosmo compress are almost identical in appearance. It thus makes sense to cover them both in a single chapter. In a single Jaleo configuration, you never will have both applications at the same time. If you have a Jaleo PLUS system, you will use the Sirius application, while you can use the Cosmo application if you have an Indigo2 or Indy system with Indy Video / Cosmo or Galileo Video / Cosmo.

Important Note: *Never* run the realtime IO applications at the same time as any other Jaleo application. Doing so is potentially dangerous for raw partition content.

2.1 Principles of Operation

The realtime capture applications allow you to use VTR remote control to shuttle through a tape, searching for sequences you want to capture. You can now mark an in and out point on the fly, or you can specify time code values directly numerically in case you have a printed list of time code values. You can now preview the capture, to make sure that you marked proper in and out positions.

As a further step, you can specify a name for the clip to be created by the capture application and you can enter a tape number.

Once all this information is specified, you can add the job to a job list. The job can be executed immediately using the Do This button, but typically you will want to add more jobs to the list and then do an automatic capture session while you go and have a coffee. A job list can also be saved and later reloaded.

With a joblist present, you press the Perform button to start capturing. All jobs in the job list will now be processed, causing the VTR to be positioned in front of the in point with appropriate preroll and started. The application will then capture the video between the in and outpoint, writing it to disk either compressed (Cosmo) or as uncompressed full resolution D1 video (Sirius). After the full res source material is captured, the system will automatically create preview images from the capture result and it will create a clip file in the current Jaleo project CLIPIMAGE directory. This process will proceed until the job list is empty.

For playout of rendered sequences, operation is almost identical. Instead of shuttling through a tape, you drag a clip file to the clip name text field to set a job. Then you set an inpoint on your VTR and you are all set. Again, you can set any number of jobs and then activate playout to a given in point on the VTR.

2.2 Starting Up the Realtime Video Application

You can start up the realtime video application, provided you have appropriate hardware, by

- selecting Realtime Video from the Jaleo Toolchest Menu or by
- typing `rtvideo` in any shell of a Jaleo configured user.

Important Note: *Never* run the realtime IO applications at the same time as any other Jaleo application. Doing so is potentially dangerous for raw partition content.

2.3 Application Usage

The application window of the realtime IO applications looks like this:

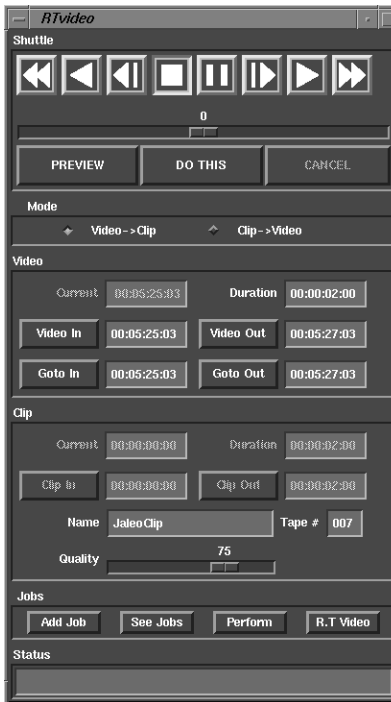


FIGURE 1. The Cosmo Realtime Application

The window shown actually is the Cosmo application, because it has a single user interface element more, and that is a slider to control compression quality.

The application consists of six separate areas:

- Shuttle Control. Remote control for your VTR.
- Immediate Execution Buttons
- Direction of Operation. Here you set up if you want to capture or playout clips.
- Video In/Out point controls. Here you specify the in and out points for a job.
- Clip Data. Here you can specify the clip information necessary for capturing a clip. In playout mode, you use this area to specify the clip to be played out.
- Job Control. Buttons to add jobs, access the job list, execute the job list and to open a reference video window.

Aside of these fields in the main application window, there are two more windows you can open:

- The Job List window, allowing you to see, edit, load and save jobs.
- The reference window, displaying the video information from the video hardware on screen.

2.4 Realtime Application Main Window Controls

2.4.1 Shuttle Control

The real time IO interface permits to remote control a connected VTR with standard shuttle buttons for Play, Fast Forward, Fast Backward, Pause, Stop, Step Forward/Backward one frame etc.

You can use the slider under the control button line like a jog/shuttle wheel: The further you push it to the left or the right, the faster your VTR will transport forward or backwards in shuttle mode. When you let go the mouse button while dragging the slider, the slider will move back to the zero position just like a spring loaded shuttle wheel.

2.4.2 Immediate Execution Buttons

The three buttons under the slider permit you to execute an operation once it is programmed.

- After setting an in and outpoint (see below) you can start a preview of the job you programmed, in effect running the VTR and the video board without actually grabbing data to disk.
- You can execute the current job after a preview was satisfying by pressing “Do This”. The settings required depend on the operation mode. Depending on if you are grabbing or playing out material, these are the minimal settings you have to make:

To create a new clip, you have to set at least:

- Video In Point to specify where the capture material comes from
- Video Out Point or Duration to define the number of frames to be captured
- Clip Name to specify under which name the new clip is to be created
- Optionally, you can add a tape id.

To play out an existing clip, you have to set at least

- Clip Name to tell the system which clip you want to play out. The easiest way to fill this field is using drag&drop from a filemanager window or from the Loader.
 - Video In Point. The position of the tape where the video will be recorded.
 - Optionally, you can use the Clip Inpoint/Outpoint or Clip Inpoint/Duration fields to define a subsection of the frames available in the clip file to be played out. By default, the Clip Inpoint is set to 0, i.e. the first frame of the clip, and the duration is set to comprise all available frames.
- Finally, you can abort any operation in progress by pressing “Cancel”.

2.4.3 Mode (Direction) of Operation

This radio button allows you to select the desired direction of operation. You can either select grab or play out mode.

- Click on Video->Clip for grabbing clips
- Click on Clip->Video for playing out clips to video.

2.4.4 In/Out Point Control

In this area, you set up the in and out points for a job. For grabbing clips, both in and out points on the VTR must be set, while for playing out a clip specification of an inpoint is sufficient. The fields in particular are:

Current

The current tape transport position.

Video In / Video Out / Duration

Two pairs of buttons/position display fields for in and out point, and a single text field for duration.

The buttons, when pressed, copy the current transport position into the in or out text field. To set an in or out point it is thus sufficient to position the transport appropriately and to press the Video In or the Video Out button.

Note: You can also press the In / Out buttons while the transport is playing. The current transport position is then read into the field.

Also all text fields are directly editable. You can use keyboard editing to set or modify values.

Instead of specifying a Video Out value explicitly, you can also modify the duration text field.

In playout mode, only the Video In and Goto In options are active: For playing out to a VTR it is only necessary to specify the Tape In Point, i.e. the position on the VTR the clip is supposed to go to, as all other data depend on the clip.

Goto In / Out

The Goto In and Goto Out buttons allow you to position the tape transport quickly at the currently selected in and out position.

2.4.5 Clip Data

The Clip Data area shows a number of information fields pertaining to a clip file. In grab mode, only the name and tape-# fields are active, as all other informations can (and will) be deduced from the Video In/Out fields. The video in and out points are stored in a newly created clip as source timecode information values.

When creating a new clip, you have to specify a clipname, while the tape id field is optional. Note that the clip description file is always stored in the CLIPIMAGES directory of the currently selected Jaleo project.

In Playout mode, you have to specify a clipname in the Clipname field before you can create a job. The easiest way to enter a clipname is to use drag&drop of a clip from either the SGI filemanager or a Jaleo application, for example the Loader. The Clip Inpoint/Outpoint or Clip Inpoint/Duration fields can then be used to limit playout to a part of the clip. Note: The Clip Inpoint is *not* the inpoint location on the VTR (this is set using the Video In field) but it describes the first frame of the clip that is to be played out. It can thus never be larger than the clip duration. The Clip Outpoint is correspondingly the last frame of the clip to be used.

Clip Image Data Storage Locations

The clip image data location depends on the system:

- For Jaleo Plus, the image data is always stored on the raw partition, i.e. the realtime capable disk array connected to your ONYX. This applies to both preview images and full res frames.
- For Jaleo Composite, the image data is always stored to a file system, as Movie files can not be stored on raw partitions. This applies to both preview data and full res compressed images.
- The Clip description files, i.e. the files with the extension .nclp that you drag into the reel for arrangement, are always stored in the CLIPIMAGES directory of the current project.

Compression Quality Control

The Cosmo version of the realtime IO application has an additional control slider in the clip area that is used to set compression quality. The higher the value, the higher the quality and the larger the images generated.

2.4.6 Job Control

The job control line of buttons contains the following functions:

Add Job

Adds the current settings as a job to the job list. If there are invalid settings, or if you have not set any required field at all, an error dialogue will appear.

To create a new clip, you have to set at least:

- Video In Point to specify where the capture material comes from
- Video Out Point or Duration to define the number of frames to be captured
- Clip Name to specify under which name the new clip is to be created
- Optionally, you can add a tape id.

To play out an existing clip, you have to set at least

- Clip Name to tell the system which clip you want to play out. The easiest way to fill this field is using drag&drop from a filemanager window or from the Loader.
- Video In Point. The position of the tape where the video will be recorded.
- Optionally, you can use the Clip Inpoint/Outpoint or Clip Inpoint/Duration fields to define a subsection of the frames available in the clip file to be played out. By default, the Clip Inpoint is set to 0, i.e. the first frame of the clip, and the duration is set to comprise all available frames.

See Jobs

Opens the job list window. Here you can modify or delete jobs, as well as load and save the job list.

Perform

Executes the job list. The list will be checked for errors before any job is executed.

RT Video

Opens a video reference window that shows you the video input of your video hardware.

2.5 The Job List Window

The job list window consists of a large text window in that you can edit your job entries as text. You can move the text cursor using the arrow keys or the mouse (click in the window at the desired edit position), and you can change the text as in any text editor by using backspace and normal letter typing. You can also select single or multiple lines and delete them.

Using the Load and Save buttons, you can write your current job list to a file and reload it to the job list window at a later time.

The Close button will close the job list window.

2.6 Remote Control Setup for Jaleo PLUS

To achieve remote control of your VTR from a Jaleo PLUS system, you need a VLAN receiver set to node number 1 that is connected to the VLAN transmitter in your Sirius Video system and to the remote control connector of your VTR. The setup looks like this:

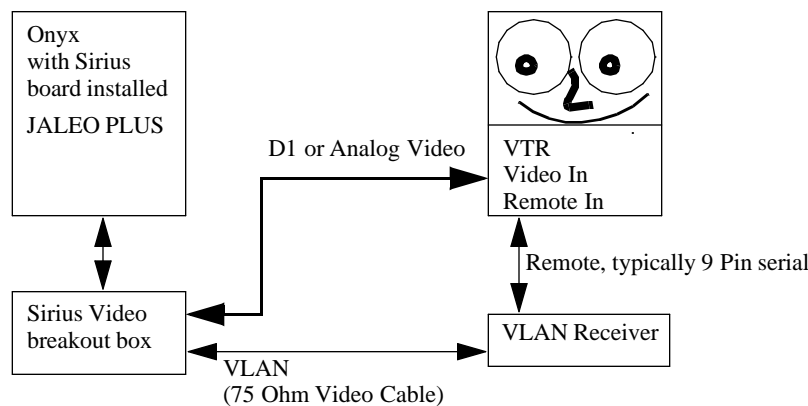


FIGURE 2. Remote Control Setup for Jaleo PLUS

There are various suitable models of VLAN receivers that can be used. The original models are little black boxes called VLAN-R, but newer systems that contain at least one externally accessible receiver like Alix or VLX should also be suitable. Contact your dealer or distributor for more information.

2.7 Remote Control Setup for Jaleo Composite

With Jaleo Composite, you need a VLAN configuration that can be controlled via a serial interface. The setup looks like this:

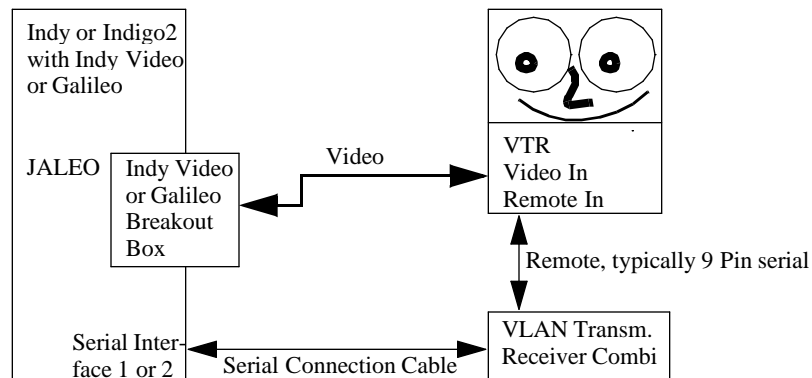


FIGURE 3. Remote Control Setup for Jaleo Composite

VLAN models suitable for this type of connection must have a VLAN Transmitter and a VLAN receiver. This could either be an Alix or VLAN Express system, or a combination of the separate VLAN-T and VLAN-R Transmitter/Receiver boxes.

2.8 Configuration Files for the Realtime IO Applications

The Realtime IO applications can be configured using two configuration files that are located in the `JALEO-ENV/etc/devices` directory. Editing of Jaleo configuration files and their syntax is described in the installation and setup manual.

2.8.1 RtVideo.cfg for Jaleo PLUS

The `RtVideo.cfg` configuration file for Jaleo PLUS contains only a single configuration parameter.

An example file for Jaleo PLUS looks like this:

```
/* Config for Jaleo PLUS */  
BACKLOAD 25
```

The `BACKLOAD` parameter is used to optimize performance for the creation of preview images. A backload value of 25, the default, means that 25 full res images are read into memory at the same time, and the scaled images for preview are saved in blocks of this size too. Reading and writing in blocks as large as possible improves performance, but requires a lot of memory. For example, 25 images require up to 42 Megabyte of memory, not counting the previews to be created and Sirius video buffers that also can be of considerable size. Specifying too large a value that exceeds the amount of available memory will slow down performance considerably due to swapping.

2.8.2 RtVideo.cfg for Jaleo Composite

The `RtVideo.cfg` configuration file for Jaleo Composite contains two configuration parameters.

An example file looks like this:

```
/* Config for Jaleo Composite */  
LOWRESOLUTIONUNCOMPRESSED  
CORRECTION 16
```

The LOWRESOLUTION parameter determines the resolution for the preview images. For proper performance in the reel window, we currently recommend a value of UNCOMPRESSED. Alternatively, a value of JPEG is possible.

The CORRECTION parameter is used to correct grab inaccuracies of the Cosmo/Galileo combination. Normally, a value of 16 produces accurate grabs. If you have deviations in either direction, modify the value of the CORRECTION parameter appropriately. If the captured sequences start to late, use a smaller value, if sequences start to early use a larger value.

2.8.3 VLAN.cfg - VLAN Configuration File

The VLAN configuration file VLAN.cfg allows you to configure the VLAN (Video Local Area Network) system used to control external video systems.

An example file looks like this:

```
CONNECTIONVLAN_SERIAL  
PORT/dev/ttyd1
```

The CONNECTION parameter describes the type of connection used for the VLAN equipment. Possible values are VLAN_SERIAL, required for Jaleo Composite, and VLAN_SIRIUS, required for Jaleo PLUS.

The PORT parameter does not have any function for Jaleo PLUS; for Jaleo Composite it is the serial connection used to connect your VLAN equipment. Possible values for this parameter on Indy and Indigo2 workstations are

- /dev/ttyd1
- /dev/ttyd2

Note that it is necessary to have exclusive usage rights and the right permissions for these serial ports. To obtain exclusive usage rights,

- Open the System Manager from the System menu of the toolchest
- On IRIX 5.3, select System Admin Tools from the Tools menu of the System Manager. Under IRIX 5.2, this step is not necessary.
- From the tools panel at the bottom of the window, select the Port Setup tool by double clicking it.
- In the Port Setup window, click on the port you want to use for VLAN. If it shows the label “Available”, you are done and you can quit the System Manager. If not, select the “Disconnect” button to make the port available or choose another port that already is available.

To grant yourself access rights to the serial interface,

- Open a shell and type

```
login root
```

If you have set a root password, you may be prompted for it.

- Type

```
cd /dev
```

to change to the directory where the UNIX device drivers are located.

- Type

```
chmod ugo+rw ttydX
```

replacing the X in the previous line with the number of the serial line you want to use.

3. Jaleo Backup and Restore

The Jaleo Backup and Restore Applications allow you to save a full project or parts of it to a backup medium and to restore the data at a later point in time.

The backup medium can be

- A normal tape drive, that is, a DAT, EXABYTE or DLT drive connected to the SCSI bus. This mode is available for Jaleo PLUS and Jaleo Composite. Using a DLT drive is highly recommended, as these drives not only have a far higher capacity (20 to 40 GB) than a DAT drive (2 to 4 GB), but they are also much faster (depending on the machine type, we have measured up to 3.5 MB per second, compared to 180 KB for a DAT). DLT drives are not much more expensive than DAT drives. Note that third party DLT drives often require a small IRIX system configuration adaptation to perform with suitable speed on SGI systems. Ask your system vendor for more information.
- A combination of a normal tape drive (see above) with a D1 or BetaCam VTR connected to the SIRIUS video board of a Jaleo PLUS system. In this mode, video sequences (i.e. the image data) from the raw partition of the Jaleo PLUS system can be backed up in real time on the VTR, while the small additional description files (Clips, Environments, etc) are saved to the normal tape drive.

3.1 The Backup and Restore Applications

The backup and restore application have practically the same user interface. As an example, here we show a screenshot of the backup application:

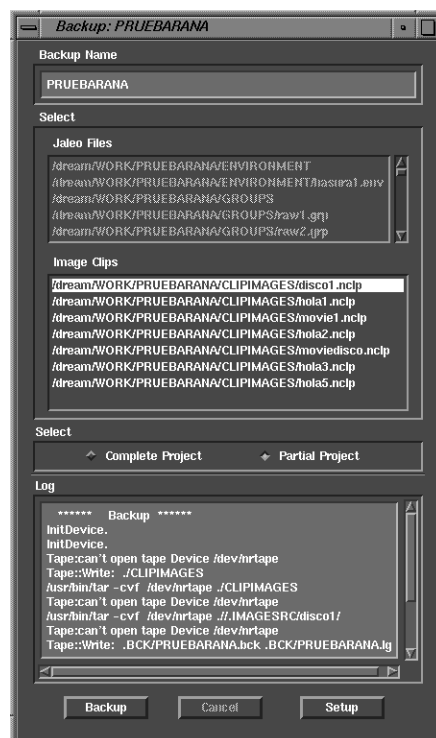


FIGURE 4. The Backup Application Window

You can start the backup or restore applications by either:

- Selecting the appropriate entry from the Jaleo menu of the toolchest.
- Typing `jbackup` or `jrestore` in any shell window of a Jaleo-configured account.

3.2 Principles of Operation

The backup and restore applications operate on the current project, as selected in the Jaleo Project Manager.

They allow to save either the complete project content or a subset of it to tape media. The choice of subsets is the first interesting one. Jaleo manages two types of data: Small description files that for example describe:

- Your compositions, or environments in Jaleo jargon.
- Saved groups.
- Saved effects.
- Material to be used. Each set of material to be used in Jaleo always is described by a clip file that contains the storage location of both the full resolution and preview size material for the given clip.

These files are small textfiles; if an environment file in Jaleo has 50 KBytes, it can already be considered big.

On the other hand, there is the image material that potentially can be huge - a second of uncompressed CCIR-601 material can take up 42 MByte.

Because the description files are so small, it does cost almost no amount of time to save all of them whenever a backup is made.

However, the backup application lets you choose which clip files you want to backup. And the choice of clipfiles does than not only include the description files, but the possibly huge chunks of material associated with it.

In short, a Jaleo backup always consists of all the data files of a project that are not clip files, and the clip files plus the associated material that you have chosen at backup time.

Because backups of material as large as uncompressed video can be very slow on normal data drives, Jaleo PLUS includes the option to use the Sirius video to write CCIR-601 images in realtime to a VTR. The backup then only stores the description files and the image positions on the VTR on digital tape, and the images themselves will go to video, providing much faster access.

3.3 Backup and Restore Details

Both applications show in their title bar the currently active project name. The project name is the name that has been activated using the Project Manager application.

The first window from the top down is the backup name. By default, the backup name is identical to the project name, but you can choose any backup name you like. The original project name will be stored in the backup tape.

The next two windows are lists of all the description files in the current project, and the clip files available, respectively. By default, nothing in these lists can be selected, because by default the

backup mode, as shown in the next section of the application window, is set to Complete backup. A complete backup always writes all of the description files and all clips to the backup medium. The other option is Partial Backup. If this mode is selected, the user can select any combination of clips from the clip list window (hold the <ctrl> key for discontinuous multiple selections. Still, all description files will be saved, but as these are extremely small compared to the data files, this does not take any amount of time. As always all description files are selected for backup, you can never select files from the upper list. It is provided only for reference.

The last list window in both applications is the message log. Here, all activities of the programs are listed so that progress of operations can be traced.

Finally, there are three buttons at the bottom: Backup (or Restore in the restore application), Cancel and Setup. While Cancel does the obvious thing, that is cancel any operation currently in progress, Setup requires a little bit more attention. To set up the system properly is of great importance for successful backup and restore.

3.3.1 Backup and Restore Setup

The backup and restore window shows all option if called up in the backup application; for restore, some settings can be deduced from the backup and need not be setup manually. The setup window, opened by clicking on the backup button, looks like this:

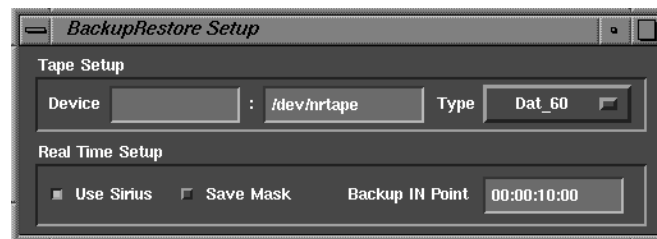


FIGURE 5. The Backup Setup Window

The setup window is used to select the appropriate devices for setup; for obvious reasons, only Jaleo PLUS shows the Real Time Setup section of the setup window, as this requires the ability to read and write fullres video in real time.

Tape Setup

The Tape Setup lets you select a tape device and a type. The tape device section is divided in a hostname section, by default empty, and a device driver section, by default the standard tape driver /dev/nrtape. The nr in nrtape stands for non-rewind. Whenever you configure your system to use a different tape drive (generally only necessary if you use multiple tape drives on one system, always make sure to use a non rewind tape driver, or your backups will not be usable. The defaults for the tape drive setup can be set in the main Jaleo configuration file, .jaleorc. Please see the installation and setup guide for more information.

If you want to configure the tape drive on the fly:

- The hostname section needs only be used if you want to use a remote tape drive, i.e. a tape drive connected to another machine in the same network. This is certainly not recommended, as network access to tape drives is slow. If you must need a network drive, specify in this field

the hostname of the host with the tape drive, as well as a user that has permissions to use the tape drives. In most cases, `guest` will be a suitable user name. The form of user and hostname must be like this:

`guest@myhost`

provided `guest` is the user name and `myhost` is the hostname.

- The device name, as said above, can be any valid UNIX tape device driver - just make sure to always use the non-rewind version of the driver. You can get a list of tape drives connected to an SGI system by typing `hinv` in a shell window. The `hinv` command will also give you information about SCSI controller and SCSI ID of the drive. All non-default non-rewind SCSI drivers have a form like:

`/dev/mt/tpsXdYnr`

where `X` is the SCSI controller number (0 on most systems) and `Y` is the SCSI ID of the drive.

Tape Length

The tape length option menu gives information to Jaleo on how large a backup on a single tape can be. Jaleo backup and restore applications can manage storage on multiple tapes, but it is very important that you specify a correct tape length for this feature to work properly. If you do have tape drives with tape lists not in the option menu, you can configure the list according to your backup drive. See “Tape Type Configuration File” on page 18 for more information. The tape length option is only visible in backup mode; in restore the end of tape is found easily automatically.

Note: All tapes used in a single backup **MUST** have the same length.

Real Time Setup

On Jaleo PLUS systems only. Here you have the possibility to activate clip source material backup to the Sirius video with the Sirius toggle. If the Sirius button is deactivated, all backup including the clip material goes to the digital computer tape. The Use Mask button activates alpha channel storage. Finally, the IN point parameter allows you to determine where on the video tape the first images should go.

3.3.2 Backup and Restore Sessions

Backup

To begin a backup session, you first start the project manager to select the target project for the backup. Then you start the Jaleo backup application.

You must now first decide if you want to backup all clips or just a selection of them. If your choice is a partial backup, you should activate the partial backup button under the project content lists. You can now select the clips you want to backup. To do multiple selections of clips that are not continuous in the list, hold the `<ctrl>` key while selecting.

Before starting the backup, you should check the setting in the Setup window. Normally, you will configure the tape drive defaults one time properly in the `.jaleorc` configuration file, so that

these do not require any attention. You should select a tape length matching the tapes you intend to use, or the backup may be invalid.

Also, set up the realtime parameters to match the clips. Saving alpha does not make sense when you do not have clips with alpha channels.

Now you are ready to start the Backup. While writing your tapes, the system writes messages in the log file that informs you about the state. In case of critical errors, error popup dialogues will appear.

Note that you **MUST** label your tapes when a backup is distributed over multiple digital computer tapes. The order of tapes is important for a correct restore, so you should write numbers on the tapes immediately. Again, it should be noted that we very much recommend to use high capacity and high speed digital tape drives, i.e. if possible by any means DLT drives that are fast and for many setups will eliminate the need for multiple tape backups entirely.

Restore

When you start the restore application, you will first be asked to enter a backup name. As you in most cases will not remember the name of the backup simply press return. The system will ask for the last backup tape and it will read a backup description from this tape. In case you do remember the backup name, you can enter it in the list. This will speed up retrieval a little bit, as the computer keeps track of backups made on one system using description files stored in your WORK directory. The directory where the files are stored is called .BCK. Normally, you will not want to touch these files. They are NOT absolutely necessary for working retrievals, but they help to gain access speed.

If the tape you enter in the backup data drive and the descriptive informations stored by the computer do not match, the computer will ask you if you want to cancel or use the information from the backup tape.

Once you have selected a backup, the computer will open the restore application window, that is totally identical to the backup application. Only now, you can not edit the field for the backup name. Instead, you see the name of the backup on tape displayed, together with the backup date and the project name of the original project you did back up.

You can now restore everything in Complete Restore mode, or you can create a Partial Restore. As always, partial restores restore all the description files, but only the selected clips.

In the setup window for restore, you only need to specify the tape drive; all other options can be deduced from the backup directory on the tape.

Target Projects for Restore

The rules for data placement with Restore are as follows:

- If you do a Complete Restore and no project with the name of the original backup project exists, the project will be created and automatically selected. All data will be restored into this new project.
- If you do a Complete Restore and there is already a project with the respective name, this project is selected. Data will be restored in this project. If files already exist with the same names, the computer will prompt you if you want to overwrite files.

- If you do a Partial Restore, your data will be restored to the current project. For image data to be stored on raw devices, this makes perfect sense. Image clips stored on file systems, however, can currently not be used automatically when restored to a different project name. Please see the description on the clip file format for information on how to adapt clip files manually to work in a different project directory (you must change the project base name in the clip file using a text editor).
- A workaround for this problem is to first create a project with the same name as the original project, to select this project and to then do the partial restore. As partial restores always go to the current project, this will produce usable clips. Again, the limitation does only apply to clips that store their image data on file systems; image data on raw devices can be restored to any project.

3.3.3 Tape Type Configuration File

The tape length option menu can be configured using the file `TapeLengthDev.cfg` in the directory `JALEO-ENV/etc/devices`. This file contains a simple list, whose entries are single lines, each with a name for the tape type and a storage capacity for this tape type. The tape names are not allowed to have spaces and special characters. Storage capacity must be set apart from the tape name by white spaces (i.e. space characters or tabs)

4. File Management Tools

The File Management Tools fill in some important gaps in Jaleo data management:

- While deletion of single files from the file system is easy using the desktop tools and shell commands, it is sometimes difficult or at least quite a bit of work to delete a set of files that together make up a single logical entity. For example, Jaleo clip files reference files containing two different versions of the actual image data and it makes life considerably easier to provide a way to delete a clip file together with these referenced source material files. As the same source material may also be used by different clip files, environments or groups, a list of material references, as provided by the `dustbin` application, helps a lot to maintain healthy Jaleo projects.
- While data on raw devices can easily be created using the IO program and the realtime applications, there is no way to delete material from a raw device with these programs. The `dustbin` application also takes care of this problem.
- The raw partition lister tool provides simple access to raw device status information and gives a listing of the raw device content, in a manner similar to the desktop tools.

4.1 The Dustbin Application

The `dustbin` application is a specialized “dumpster” version, created to deal with Jaleo clip files. It operates just like the normal trash can of the SGI desktop, by using drag&drop. You can drop any file on the dustbin desktop icon, however, if the file is not a Jaleo clip file, the `dustbin` application will only ask you if you indeed want to delete the file. If you answer positively, the file will be lost.

The main window of `dustbin` looks like this:

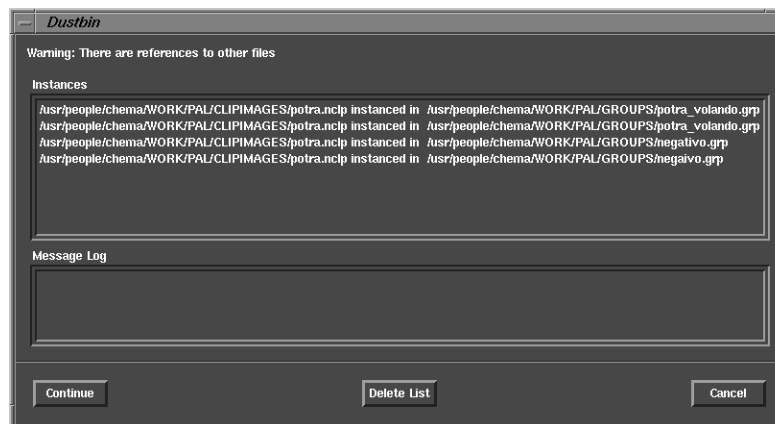


FIGURE 6. The Dustbin Application Window

The normal way to start the `dustbin` application is to drop a (clip) file on its icon. However, you can also start `dustbin` by double clicking on the icon or by typing `dustbin` in any UNIX shell window of a Jaleo user. If you start the application by double clicking or from a shell, a small dialogue is displayed in that you can enter a file name to delete. You can use drag&drop to place a filename in this dialogue as well. Simply drag the file you want to delete from an SGI file

manager window or from a Jaleo application on the dustbin dialogue text field where you are asked to enter a clip name.

4.1.1 The Dustbin and Clip Files

Special treatment is offered for clip files. If the file to delete is a Jaleo clip file, the `dustbin` will quickly search through all files in all projects in the Jaleo work directory. It will then present you with a reference list for the file you try to delete, showing you all files that make use of this clip file, or of the same material that is used by this file. You will see all environments or groups that make use of the material used in the clip, and you will see all other clip files that make use of the same material.

Also, it is normally cumbersome to delete a clip with all its associated data. As a clip file references preview and full res data that can be stored in different storage locations and different formats, even in storage locations that are not accessible to normal desktop tools, it is a great help to have a tool that permits you to delete the clip including the referenced materials in a single step.

And the bitter decision to purge precious material irrevocably from your disk is becoming a bit easier, as the cross reference list makes sure that you know exactly which other files need the material you are about to delete. The `dustbin` application also allows you to delete all files that reference a particular set of materials with a single button, so that you can effectively get rid of a whole project at once.

If you are not familiar with the Jaleo clip file concept, please make sure to read the introductory chapter of the Users Manual.

4.1.2 The Dustbin and Raw Partition Storage

Clips stored on raw partitions do not need any special treatment with the `dustbin` application. As it is made to deal with clip files, it handles deletion of material independent of the actual storage location of the clip.

4.1.3 Operation and Deletion Options

After dropping one or more clip files on the `dustbin` icon on the SGI desktop, the system does the cross-referencing search and then presents the application window displayed above. You should now check the list to see if any of the files that make use of the material is still of any value to you. If so, you should quit the `dustbin` application and keep all the files. If not, you have two choices:


- You can delete only the clip files you dropped on the dustbin icon and the material referenced by them. All other files that use the material are kept. However, these files will become at least partially useless, as material referenced by them will be deleted and not accessible any more. In some cases you probably will not mind, for example if you intend to replace a certain clip used in a reel by another one anyway, and you first need the space to store the new clip. This deletion mode is selected by clicking on the “Delete Instances Only” button.
- You can also delete all files that reference the same source data. This, of course, is a potentially dangerous decision, as it may remove a great number of files from your system. Please read the full cross reference list in the window before you select the “Delete Instances and other Users” button.

4.1.4 Deleting Clip Files without Material

If you only want to delete clip files without deleting the associated material, use the SGI dumpster application/desktop icon or use the shell commands. Note that the SGI dumpster has a mode that allows you to keep deleted files in a special directory until you explicitly empty the dumpster. This can potentially fill your disk drives very rapidly if you delete image files that way. Empty the dumpster regularly.

4.2 The Raw Device Lister Application

The Raw Device Lister Application, `xlsRaw`, gives you a list of the current raw device content. Of course, this requires that you have a raw device configured on your Jaleo system. As this is highly recommended, you will use `xlsRaw` sooner or later.



The screenshot shows the `xlsRaw` application window. It has a title bar with the text `IsRaw`. Below the title bar is a section titled "Raw Device Content:" containing a table with four columns: `ClipName`, `NoImg`, `BlockSize`, and `PixelSize`. The table lists several entries with their respective values. Below the table, there are four rows of statistics: "Files on Raw Dev:", "Used (Mb & Sec):", "Free (Mb & Sec):", and "WORK:". Each row has two input fields. At the bottom, there are two buttons: "Quit" and "Update".

ClipName	NoImg	BlockSize	PixelSize
/PAL/L_vtr1	3	2633	4*180*144
/PAL/H_vtr1	1	2836	3*720*576
/PAL/L_cosa	59	5266	4*180*144
/PAL/L_potra_volando	48	25566	3*180*144
/PAL/L_potra	48	36098	4*180*144
/PAL/H_potra	48	45842	3*720*576
/PAL/H_potra_volando	48	162482	3*720*576
/PAL/L_bichos	68	279122	4*180*144
/PAL/H_bichos	68	292926	4*720*576
/PAL/H_cosa	59	513246	3*720*576
/PAL/L_anc1a	101	656616	4*180*144
/PAL/L_tarari	100	677119	4*180*144

Files on Raw Dev:	14	WORK:	7653376
Used (Mb & Sec):	4255	Used (Mb):	7635968
Free (Mb & Sec):		Free (Mb):	17488

Quit	Update
------	--------

FIGURE 7. The Raw Device Content Lister

The `xlsRaw` window shows you a list of the image material stored on the raw device in the scrolled window and offers some statistics on the raw device totals, as well as the file system partition that holds your Jaleo work directory. Finally you have a button to update the listing and one to quit the application.

4.2.1 Start Up

To start `xlsRaw`, you can either

- select the appropriate entry in the Jaleo toolchest menu, or
- type `xlsRaw` in any shell window

4.2.2 Details of the Listing

File List

The file list contains the following items for each content item stored on the raw disk:

- Image format, given as horizontal and vertical resolution and the number of bytes per pixel. The number of bytes per pixel typically is 3 or 4, for images without or with alpha.

- Sequence name
- Number of images
- Size of the sequence in MByte

Statistics

The statistics section gives you the total number of clip files on the raw disk, as well as used and free space, both in bytes and in seconds. Note that the seconds values are estimates. To convert from bytes to seconds, the current settings from the `.jaleorc` configuration file are used, in particular image size and frames per second.

In the third column, the total size, amount used and amount free of the disk partition that contains the Jaleo work directory is shown.

5. Command Line Tools for Raw Partitions

There are three command line tools for use with raw partitions:

- `lsRaw`. A raw device lister to be used in a shell
- `rmRaw`. To delete an image sequence stored on raw devices
- `compressRaw`. To compact raw device storage

5.1 Listing Raw Device Contents

The `lsRaw` command is used without arguments. Simply type

```
lsRaw
```

in a shell window. A listing very similar to the one presented in the graphical listing application will be presented.

5.2 Removing Sequences From Raw Devices

To delete a sequence stored on a raw device, type

```
rmRaw sequenceName
```

in a shell window, where `sequenceName` is the name of an image sequence that lives on the raw partition. You can get a list of the sequence names using `lsRaw`. Note that using the `rmRaw` program potentially leaves clip files that try to reference images not existing any more. Therefore, we highly recommend to use the `dustbin` application for raw device cleanup instead.

5.3 Compacting Raw Partitions

Although Jaleo normally takes care to prevent fragmentation of the raw partition. However, you can compact the raw partition manually by typing

```
compressRaw
```

in a shell window. Note: You must *never* run `compressRaw` while any other Jaleo application is running.

6. License Management Tools

Jaleo versions after (and including) version 2.1 have a new, very flexible licensing scheme. Instead of using a proprietary license handler, Jaleo applications now use a standard license management system. For you, on the user side, there are a number of important advantages, the most notable of them being that you can use your software on any machine connected to a network instead of deciding at purchase time for a particular CPU.

The license management of Jaleo is based on the Élan License Management system, (c) 1989-1994 Elan Computer Group, Inc. All rights reserved. Élan License Manager is a trademark of Elan Computer Group Inc.

6.1 Introduction

The Élan License Management system is a network based license manager. Systems of this type work by running a license server on any machine of the network (and can work on a local machine just as well as on any other machine). The license server has all informations about licensed “features”, i.e. of the parts of a product for that you have purchased a license, or a “right to use”. In Jaleo for example, the main application, rendering, IO, the paint system etc. are all separate features that can be activated independently.

Whenever a Jaleo application is started on any machine on the network, it first contacts the license server and requests a license for the feature or features the application provides to the user. Depending on the current status, the license server may grant a license or reject the request. If you for example have 3 Jaleo licenses but five machines, you can run the software on any 3 machines at the same time, but when you try to run the software a fourth time, it will refuse to operate, as the license server will not grant another license. When you terminate the software, it will actually return the license to the license server, so that it can be used by another client.

As these type of licenses are independent from the actual machine the software runs on (the only constraint is that a license server must be reachable inside of the network), they are also called “floating licenses”.

If during execution of an application that requires a license the license server becomes unavailable, the application will become inoperable after a grace period. It will always allow you, though, to save your work before it does so.

Floating licenses managed by a separate license server give you a lot of flexibility, as you do not have to lock software to a particular machine. Instead, a license is locked to a license server that negotiates distribution of features to your workstations.

6.1.1 Shared Licenses and License Hold Period

Normally, each instance of an application requires a separate license. As this behaviour does not make a lot of sense for Jaleo, where multiple instances of the same or different applications may be running, we are using shared licenses. That is, for all instances of a given feature that are running on the same computer, started by the same user and displayed on the same computer monitor, only a single license is required.

This behaviour also ensures that software can be re-run immediately after a crash of an application - in a crash situation, the software can not always return its license safely to the license

server. Although the license server will recognize the failure of the client after a certain period, you certainly do not want to wait 2 or three minutes before you can proceed with your work.

Even if a license has not been returned properly, you can run the software again on the same computer, because shared licenses permit you to have as many active copies of an applications as you wish.

Another problem in a multi user environment is the danger of “feature robbery”. If you quit your application for a very short period for another task, you want to be able to restart your work after a brief period without running in danger that another user grabs your license. This can be achieved with so called “Held Licenses”. A held license is a license that is hold for a particular user a short amount of time after the application has been terminated. It is immediately accessible, but only to the user who just returned it. The period of time licenses are held can be controlled by the user. See the section on `elm_resource` for more information.

6.1.2 Manual Interaction

In some situations, manual interaction with the licensing system is desirable. One reason to do so is license input, and another monitoring the licensing activities.

Another example is that after a hardware failure on one machine you want to go on working immediately, but on another machine. In this case you may want to inform the license server manually that a license is to be returned. Should the application that uses the given license still be running, it will be terminated by this activity. Licenses can only be terminated by the user who “owns” the license server process or the user who owns the process that requested the license. This prevents other users from terminating your licenses. The command to use for this purpose is `jlicadmin -b`.

With the licensing management system, there are a number of user accessible administration programs that are available to you to manage your account.

6.2 Applications

The applications covered in this chapter are part of the Élan License Management system. Installation of the Élan software is part of the normal Jaleo setup process and is thus covered in the installation and setup manual.

This section will give a brief summary of the available commands. Reference information on each of them follows.

6.2.1 Application Summary

The most important application of the licensing software is the license daemon. The Jaleo license daemon is called `jaleolicd`. You usually never have to run it manually, as the Jaleo installation will take care of placing it in your system start-up script. There always must be one copy `jaleolicd` running on your designated license server, as Jaleo will not run if it can not receive a license from a license server.

The second most important application is called `jlicadmin`. Using this program, you can get manage your license system. From entering a license code to checking license availability up to terminating licenses of hung applications, `jlicadmin` performs it all.

There are more tools to give you reports on your license usage (`jlicreport`), to check license expiration dates (`jlicalert`), to continuously monitor license status (`jlicusage`) and to retrieve Élan version codes from Jaleo executables (`jlicver`). These tools will be very rarely used in typical Jaleo installation and are provided for the sake of completeness only.

All license management applications are located in the directory `JALEO-ENV/extern/license` in any account prepared for Jaleo. The license daemon application is located in `JALEO-ENV/extern/license/daemon`. You can execute the applications from a shell window in any account prepared for use with Jaleo.

6.2.2 License Daemon (`jaleolicd`)

The license daemon is a program that almost never is used explicitly. The Jaleo installation takes care of installing the daemon on your designated license server so that it will be run automatically when you boot the system. Normally, you will not have to touch it. There are some additional configuration options that are described in the sections on `jaleolicd` and `elm_resource` below.

You can test for presence of a daemon in the net with the following methods:

- Use the `jlicadmin` command with the `-l` option on any machine of the network. You will be presented with a list of all licenses managed using the Élan license manager that are available in the network, and of the servers that manage them. Make sure your Jaleo licenses are visible.
- On the license server itself, you can enter the command
`ps -e | fgrep jaleolic`
Do *not* type the full daemon name (`jaleolicd`) after the `fgrep` command - if the name to look for is longer than 8 characters, it will not be found. If this command returns without printing out a number and a name, no server is running. See the section on troubleshooting for more information on running servers manually.

6.2.3 Adding Licenses with `AddLicense`

The process of adding licenses is described excessively in the installation manual. It should, at this place, just be noticed, that the `addLicense` tool is the most comfortable way to add Jaleo licenses to your system.

6.2.4 License Administration with(`jlicadmin`)

The `jlicadmin` program is used for license administration. It can be used to enter licenses, to retrieve informations about licenses present in the network, to terminate the license server daemon, to manually terminate any license currently in use (`jlicadmin -b`) and for many more operations. Please see the reference page on `jlicadmin` for more information.

6.2.5 License Alerts with (`jlicalert`)

`jlicalert` can check for the expiry of time limited licenses. As Jaleo will print out appropriate messages at start-up time, you will probably not use it too often. It is also of more interest in large Élan maintained installations.

6.2.6 License Reports with (jlicreport)

The `jlicreport` tool accesses the license server log and creates usage reports. It does, however, not make too much sense in situations with only a few licenses. It is mostly intended for large Élan-managed installations and is supplied to you mainly for the sake of completeness.

6.2.7 License Monitoring with (jlicusage)

While `jlicadmin -l` prints out the current license status once, `jlicusage` monitors licenses continuously.

6.2.8 License System Version: (jlicver)

The `jlicver` program prints out the license management software release of any executable built with Élan. To use it, type `jlicver <programname>`. Note that the program name must be a full path. For example, for `jaleo` you could type:

```
jlicver ~/JALEO-ENV/bin/jaleo
```

6.3 License Troubleshooting

There are a number of possible error sources in case you do not receive any licenses. The following section will guide you through license troubleshooting.

6.3.1 No More Licenses

If you receive this message when starting up a Jaleo application, then the license daemon is convinced that all licenses you have purchased are currently in use. In this case, you should do the following steps:

- First check that the licenses you own for the given feature are not granted to other users. Run `jlicadmin -l` to see the current license usage. If all licenses are used by users from other machines, you have to check with these users to leave the application. If the entries in the listing from `jlicadmin -l` do not correspond to applications actually in use, you can use `jlicadmin -b <CID>` to free these licenses manually. The value `<CID>` is the client id printed out by `jlicadmin -l` in the first column of the license usage listing.
Note: you can not run the `jlicadmin` command with the `-b` option (for “bury”) unless you are logged in as root. You do, however, not need to log out of the Jaleo account for this purpose. You can use the `su` command in any shell to give yourself root permissions temporarily (provided you know the root password). You can also become root in a single shell window a bit more persistently by typing `login root`. This will do a formal root login inside of the shell window.

6.3.2 No Server for Requested Feature / Cannot Contact Server

If you, upon start-up of Jaleo, receive an error of this type, no server can be contacted by the application. Make sure that your license server is up and running.

- On the machine where you have your license server installed, run the command `ps`, that gives you a list of all running processes. Type

```
ps -e | fgrep jaleolic
```

Do *not* type the full daemon name (`jaleolicd`) after the `fgrep` command - if the name to look for is longer than 8 characters, it will not be found. If there is no line containing the word `jaleolic` printed, no server is running.
- Restart the server manually typing

```
/etc/init.d/jaleolicd start
```

Check for error messages after start-up.
- If the server is running and you can still not access it, you should check your network. Make sure that the target machine can reach the license server. Use the command

```
/usr/etc/ping <hostname>
```

to test if a particular host can be reached via the network.
- If the network is up, you should try to manually halt and rerun the server. Run

```
/etc/init.d/jaleolicd stop
```

and then

```
/etc/init.d/jaleolicd start
```
- Also make sure that the server is configured for autoboot. Run the program

```
chkconfig
```

to see if the flag `jaleolic` is set to “on”. If not, run

```
chkconfig jaleolic on
```
- If there are further problems, consider a server reboot. Make sure, as described before, that the server autoboot is activated.

6.3.3 Server Down

This message appears while running a Jaleo application if the connection to the license server is lost for any reason. For troubleshooting, first check if the server is running and then make sure your network is up and running.

6.3.4 Feature Expired

A temporarily licensed feature expired. Time limited features can not be used over their expiration date. If the feature expires while in use, you will have to terminate your session, although you can save your work before.

6.3.5 Access Conflicts

In case you have multiple Jaleo license servers running that maintain different license sets (a non-recommended configuration) access conflicts may arise, i.e. a client may try to contact a server that does not have a feature available. To prevent this, you can force a client to use a specific

server by setting the environment variable `JALEO_ELMHOST`. You can find more information on that in the installation and setup guide.

6.3.6 Communication Conflicts

In case a port number for license communication is used by another, non-Jaleo application as well, communication conflicts may arise. By placing an entry in both the license servers and the clients `/etc/services` communication configuration file these conflicts can be prevented. You can find more information on that in the installation and setup guide.

6.3.7 Other Errors

The license server keeps a logfile keeping track of the last ongoing transactions. The log file is maintained on your license server machine under the name `/usr/adm/elm.log`. If the content of this file does not make sense to you, please contact your dealer or distributor with this file or a printout of it (as it can be large, the last few pages will probably be sufficient) and a clear description of the problem accessible.

6.4 License Administration Reference Pages

The following information has been extracted from the Élan License Management System reference manual pages. They are reproduced with permission and the explicit copyright of Elan Computer group. Inc.

6.4.1 Élan daemon for Jaleo (`jaleolcd`)

```
jaleolcd [ -e path -f -l file -m size -p # -r file -s # -z # ]
```

Description

The `jaleolcd` command starts the jaleo Élan license manager daemon. You will not usually have to do this manually, as after a default installation the daemon is started automatically at boot time from `/etc/init.d`.

When running, `jaleolcd` will process client requests, which include issuing and returning licenses, and interfacing with `jlicadmin` to list the current users of licensed applications.

For each Élan licensed application, there are one or more key files. These files are expected to be found in the `a` directory. The default directory for the Jaleo daemon is `/usr/lib/elm/jaleo`. A different key directory or directories may be specified with the `-e` option to `jaleolcd`, described below.

Key files may be freely installed or updated while `jaleolcd` is running. Keys will automatically be reread if `jaleolcd` detects a modification date change.

The key file itself has the same name as the feature. The file is usually created with the utility script `addLicense` (see “Adding Licenses with `AddLicense`” on page 26) that internally calls `jlicadmin`. This is accomplished via a command of the form:

```
jlicadmin -c alias
```

`alias` is a human readable form for the feature number used to enumerate features internally. The `addLicense` script defines proper aliases for Jaleo features.

If an alternate key location is desired, the command would instead be

```
jlicadmin -c -e keypath alias
```

Refer to `jlicadmin` for other options.

Each key file contains the encrypted key which encodes the feature name, the maximum number of licenses available for the feature, and the feature's expiration date. The manager `jaleoliced` reads this key and will not issue more licenses than indicated or past the expiration date.

Options

- `-e path`

Optionally specify the key directory, or set of directories, and load all keys found in the directories.

Directories are specified as a colon (:) separated list of directory names. For example:

```
jaleoliced -e /local/elan:/local/express:/usr/lib/elm
```

- `-f`

Normally, `jaleoliced` will run in the background (it forks a child process, detaches from the terminal, and the parent exits). The `-f` option inhibits this, and causes `jaleoliced` to run in the foreground.

- `-l file`

Creates a log file named `file` and writes all relevant information to that file. If the file already exists, `jaleoliced` will append to the end of the file. Note that `file` must be a complete path name.

Use of a log file is highly recommended as it provides a way of tracking problems if they occur. A log file must be enabled for `jlicreport` to provide reporting information.

Please see `-m` option also, below.

- `-m size`

Limit the log file to the size specified. With this option, the log file is “self maintained” by `jaleoliced` such that `jaleoliced` limits the size that the log file may grow to. When this size is reached, the log file named `file` is moved to `file.old` and the current log file is truncated.

The size is an integer or floating point number. The default units are bytes. It may also be specified in kilobytes by suffixing a “k” or megabytes by suffixing an “m” to the size.

For efficiency, the log file size is actually checked only every 100 lines, so it might grow a bit beyond the limit before it is backed up and truncated.

- `-p #`

Cause `jaleoliced` to use the port address `#` instead of the default or one specified by the system service. One might use this option if there was a port address conflict and the user did not have permission to alter the system port services file.

- `-r file`

Issues licenses to the users according to the specifications in resource file. See `elm_resource` for an explanation of this file's format. Note that if this file is changed, `jaleoliced` will automatically reread it - it is not necessary to restart `jaleoliced`.

- `-s #`

Set the start-up time to # in seconds. This is the period that `jaleolcd` waits and listens for client reconnects after a possible crash. The default is 180 seconds (3 minutes.) Jaleo start-up files set a start-up period of 0 seconds. If you are running on a network, you may want to change the default options.

- `-z #`

Set the zombie to # seconds. If a client is not heard from within this time period, it is declared dead and its licenses are returned. The default and minimum zombie interval is 3 minutes. Note that `jaleolcd` only cleans up zombies every 60 seconds, so it may take up to 1 minute more for the licenses to be returned.

Note

The `jaleolcd` daemon is started from the `/etc/init.d/jaleolic` file. If you want to change default `jaleolcd` options, you will have to edit this file as super user. Maintain an original copy if you make any changes. After changing the options, stop the daemon using `/etc/init.d/jaleolic stop` and restart it with `/etc/init.d/jaleolic start`.

6.4.2 Jaleo License Daemon Administration Program (`jlicadmin`)

```
jlicadmin [-b cid -c -e keydir -h -i -k -l -z]
```

`jlicadmin` is a license administrator program for use with elm embedded applications.

Its functions include installing license keys, listing the users of licensed products, and other elm administrative functions.

Options

Those options marked with an asterisk (*) may only be invoked by: the owner of the current license server process or the super-user. Note that if you started the server daemon via autoboot (default for a Jaleo installation) the owner of the daemon process will also be root.

- `-b cid *`

Bury client. Return all licenses used by the client whose client ID is `cid`, and mark this client as dead. The client ID (CID) may be determined with `jlicadmin -l`.

A buried client will continue to run, but without licenses, and thus will terminate itself soon.

- `-c`

Creates a license key file. Use of this option is discouraged in Jaleo; please use the `addLicense` script instead (see “Adding Licenses with `AddLicense`” on page 26). `jlicadmin` will print a server code which the customer gives to the application vendor who, in turn, will supply a key. Note that a single application may have one or more keys, one for each feature.

If an alias appears as the final argument, then this name will be installed as an alias for this feature. One typically creates an alias for numeric features. Subsequent commands such as `jlicadmin -l` will print the alias instead of the feature name.

A list of definitions may also appear on the command line. Upon decoding the key input, the feature name gleaned from the key will be matched against the list of definitions. If a match is

found, the name following the “=” sign is taken as the feature alias. This provides a convenient way of specifying feature aliases within shell scripts. For example,

```
jlicadmin -c 1000=JaleoComp,1001=JaleoMT,1002=JaleoIO
```

will cause the alias `JaleoComp` to be selected if the key is for the feature `1000`.

Jaleo Keys will be placed in the directory `/usr/lib/elm/jaleo`. You can relocate this directory with the `-e` option, but then you must also redirect the directory used by the license daemon.

If you are using the `-c` option and you are not super user or owner of the daemon process, you will receive an error message after key installation. The error does not affect successful installation of the license. The license will immediately be accessible to a client requesting it, but it will not be visible to `jlicadmin -l` until it has been requested for the first time. The error message can be suppressed by specifying the `-p` option along with `-c`; if you want the license to be visible immediately to `jlicadmin -l`, install the license as root.

- `-e keydir`

This option is only meaningful when the create option `-c`, described above, is used. It allows you to specify a different target directory for the key to be entered. You normally will not want to use this without also changing the daemon start-up files, as Jaleo will not find its licenses otherwise.

- `-h`

When used with `-l`, produces a more expanded listing including each client's hold time and the shared license count.

The hold time is the period after the client exits that the licenses will be held (reserved) for the user. If the user starts the application again before the hold period ends, the held licenses will be assigned to him. The hold time is specified in the resource file: see `elm_resource`.

If the client is held, column one of the listing will be flagged with an H.

The shared license count is the number of clients that are currently sharing the license for the indicated feature. If the client has one or more shared licenses, column one of the listing will be flagged with an S.

- `-i`

Prints the release number of `jlicadmin` and the release of the `elm` library it was compiled with.

- `-k *`

Kill the `jaleolcd` daemon `jaleolcd`. The daemon will gracefully terminate upon receipt of the kill request.

- `-l`

Lists the current users and outstanding licenses available for all features known to all Elan license server daemons on the network. Known features are those features used since the daemon was started (usually the last time the system was rebooted.) If features are given on the command line, the listing is restricted to the features named.

The first field of the listing produced by `jlicadmin` contains a flag to indicate shared and held licenses. The next field, CID, is the client ID. Each client application will be assigned a unique ID. The CID may be used with the `-b` option described above.

The LID is the license ID. These are unique license numbers. Each client will normally have unique LIDs for each licensed feature, except when the license is shared. If clients share a license, they share the LID.

The user is displayed as `user@host,display` where `display` is the value of the X-11 `DISPLAY` when the application was invoked. The display may be missing for non X-11 applications.

The `-l` option scans all servers of the network.

- `-z *`

Clears (zeros) the log file. If `jaleolcd` is run with the `-m` option, as is default with the Jaleo start-up files, this is done automatically once in a while.

6.4.3 Alert When a License Becomes Available or Will Expire (`jlicalert`)

`jlicalert [-a -b # -e dir -f -h host -m -o percentage -p -u user -v -x days] features...`

Description

`jlicalert` provides a notification when licenses become available to you or are close to expiration. The command checks the availability/expiration of licenses, and, at your option, either sends mail or prints a message to the screen with the requested information.

`jlicalert` may be used if the licensed application does not implement license queuing, or if it is more convenient to receive notification this way. It also may be used by users or system administrators to track license expiration. For example, an administrator can request notification when licenses are within two weeks of expiration.

Normally, you will be notified when each feature mentioned on the command line is available. With the `-x` option, expiration, rather than availability, is checked. The `-a` option may be used to only alert you when all of the licenses are available or will expire.

The options `-m`, `-p`, and `-v` control what and where `jlicalert` sends its information. More than one may be used at a time. For example, use of `-p` and `-m` causes printing to the screen and mailing. The default option `-p` is assumed if none of these three are specified.

Options

- `-a`

All. If multiple features are named on the command line and the `-a` option is used, notification will only occur when all the licenses mentioned meet the requested condition are available or will expire in under the `-x` specified days. Without the `-a` option, notification will occur for each feature individually.

- `-b #`

When used with the `-f` option, check every `#` seconds, instead of the default 60 seconds.

- `-e dir`

Tell `jlicalert` that the key directory is `dir`, instead of the default `/usr/lib/elm/jaleo`.

- `-f`

Forever. Normally, `jlicalert` checks for the condition once, then exits. With `-f`, `jlicalert` runs until the condition is met.

Under the `-f` option, once you are notified about a feature, the feature is removed from the check-list. When you have been notified of all feature licenses, `jlicalert` exits.

- `-h mailhost`

Specify the host name that supports SMTP mail forwarding. If unspecified, the current host is used.

- `-m`

Mail. Mail is sent to the invoker's user name, or the name set via the `-u` option, if used. This option will only work on systems with an SMTP (Simple Mail Transfer Protocol) service. (See `-h` option above.)

- `-o #`

Occupancy monitoring.

If the number of licenses in use divided by the total number available reaches or exceeds `#%`, then you will be notified. Thus, `-o 80` warns when license usage has reached 80%. And, `-o 100` warns when no more licenses are available. If a feature has a soft limit, the soft limit will be used in the calculation.

- `-p`

Print. Information is printed to the screen (`stdout`) when the license(s) become available or will expire. This option is the default, but you may wish to use it with `-m` to indicate both printing to screen and mail.

- `-u user`

Set the recipient to `user` for the mail (`-m`) option.

- `-v`

Verbose.

Print information about the status of the license(s).

- `-x days`

Without `-x`, or `-o`, `jlicalert` will notify when licenses become available. If `-x` is used, notification is instead made if the feature will expire within the specified number of days.

Examples

Send me mail when a license for the feature `JaleoPlus` becomes available:

```
jlicalert -f -m JaleoPlus &
```

Send `scott` mail if the license for the feature `JaleoPaint` is within two weeks of expiration.

```
jlicalert -m -u scott -x 14 JaleoPaint
```

of course, the latter example only makes sense when the machine is not shut down periodically, or if you run the script from your start-up files.

6.4.4 Statistics on License Usage (jlicreport)

`jlicreport` [-d daterange -f feature -h -i -l -m# -t timeunit -u category -w #] [logfiles | logdirs]

Description

`jlicreport` produces a report of licenses available or license activity from one or more license server log files. The default log file is `/usr/adm/elm.log`

Note that `jlicreport` may be used with log files from all releases of Élan.

Please refer to `jlicadmin` and `jlicusage` if a snapshot of current license activity is desired.

`jlicreport` can display its information in two formats: numerically and as a histogram.

Time units can be specified in month, day, hour, minute, and even second.

Numeric Reports

The fields in a numeric report are:

- **Feature.** The feature alias, if one exists, otherwise the feature name.
- **Total Requests.** The total number of license requests for the Feature during the time period.
- **Total InUse.** The maximum (peak) number of licenses that were in use concurrently during the time period.
- **Over SoftLim.** If a soft limit has been applied to the feature, this will be the number of licenses in use over the soft limit during the time period. (Total InUse - SoftLimit.)
- **Number Issued.** The number of licenses successfully issued during the time period.
- **Number Denied.** The number denied, most likely from the maximum number of licenses being already in use during this time period.
- **Percent Denied.** The percentage of denied licenses out of the total number of requests for the period.
- **Total Time Used.** The total time that the feature was checked-out during the period displayed in hours:minutes:seconds.

Reports can be further detailed not only by time, but by user, `user@host`, and/or `user@host+display`.

Histogram Reports

Use of the `-h` option will produce a histogram instead of a numeric table. The same options that apply to numeric tables can be applied to report in histograms:

When displayed as a histogram, the number of successful requests is displayed as a series of pluses (“+”) and denied requests as a series of minuses (“-”), followed by the actual number of successful and denied requests, plus (+) separated (as in 54+4). The total number of requests is the sum of these two numbers.

The second line displays the peak concurrent usage. The open circles (“o”) indicate usage under the soft limit, and asterisks (“*”) indicate usage over the soft limit. This line is followed by the in

use count under the soft limit and the in use count over the soft limit, plus (+) separated (as in 26+0). The total concurrent in use is the sum of these two numbers.

Histograms are normalized for a screen width of 78, which may be adjusted with the `-w` option.

Installed Licenses

A summary of installed licenses can also be obtained via the `-l` option of `jlicreport`.

The summary will be sorted by the feature name, or alias, if one.

Options

- `-d [from - to | from]`

Limit reporting for dates between from and to inclusive. The from and to dates may be specified in one of two formats:

Feb 3 12:30:00

2/3 12:30:00

In the first format you specify the first three letters of the month. In the second form, the month number followed by a slash is used. For example:

```
jlicreport -d "Mar 15 - Apr 15 5:30"
```

```
jlicreport -d "3/15 - 4/15 5:30"
```

Note that in either format, only the month is required - the day, hour, minutes, and seconds will default to zero if missing. If the hyphen and to date are missing, the report will run to the last date.

- `-f feature`

Only list activity for the feature named. Multiple `-f` options may be used to limit display to several features.

- `-h`

Print a usage histogram instead of numeric data. (See also `-w` option below.)

- `-i`

Prints the release number of `jlicreport` and the release of the Élan library it was compiled with, then exits.

- `-l`

License listing.

Display a summary of Élan licenses available along with the number of tokens, soft limit, start date, expiration date, and key file name. No other options are meaningful with `-l`.

- `-m #[units]`

Minimum duration. With this option, commands that execute for less than the time specified by `#` will be ignored. A single letter `s`, `m`, `h`, `d`, or `w`, following the `#` may be used to specify units of seconds, minutes, hours, days, or weeks. The default is seconds.

With a low `#`, such as 5 or 10 seconds, this option may be used to effectively eliminate phantom statistics - those from commands that checks out, but fail to execute.

This option may also be used when you wish to eliminate short application sessions from your statistics. For example, `-m1m` will only display applications that have run for at least one minute. With a large number, only long application sessions will be gathered. For example, `-m1h` will only display applications that have run for at least one hour.

- `-t timeunit`

Display total license activity for each timeunit, where timeunit may be month, day, hour, minute, second. (A unique abbreviation, such as “min”, is ok.) For example:

```
jlicreport -d "3/15 - 4/15 5:30" -t hour
```

If the `-t` option is not specified, a grand total only is printed.

- `-u category`

Detail report by each user, as specified by category. The category may contain one or more key letters “u”, “h”, and “d” for user name, host, and display, respectively. Key letters may be combined in any combination of 1 to three letters. For example, `-uu` details per user name only;

`-uh` details per host name only; and

`-uuh` provides more detail,

with results displayed for each distinct user@host.

- `-w #`

Specify a display width of # character units. The default is 78. This option is currently only effective with the `-h` (histogram) option.

The remaining arguments are zero or more license server log file names or directories. If any argument is a directory, then all log files in that directory are digested.

`jlicreport` will silently skip files that are not license manager log files.

If no files are specified on the command line, then the default `/usr/adm/elm.log` will be used.

6.4.5 Continuously Report Élan License Activity (`jlicusage`)

```
jlicusage [ -b seconds -i ] [ features ... ]
```

Description

`jlicusage` produces a snapshot report of current license activity, updating every 10 seconds.

For each feature in use (or in past use) a line is printed with the current number of licenses in use, the total number of licenses available, and the current occupancy, the percentage of in-use licenses divided by the total number. A high percentage indicates that it might be wise to purchase additional licenses. If occupancy hits 100%, a series of asterisks are printed in the right column.

- `-b seconds`

Change the interval to n seconds, from the default 10.

- `-i`

Display version information.

- features

Only examine the features named, instead of all features.

6.4.6 Extract Élan Version Information From a Program File (jlicver)

`jlicver [-v] files...`

Description

`jlicver` scans the files named for the Élan version string and, if the `-v` option is used, other version information, and displays it, if found. It will be displayed in the form:

file: Elan License Manager release 3.0.6

If the `-v` option is used, a message of the following form will be printed:

file: Elan License Manager release 3.0.6

file: HAVE_SIG=SIGACTION

file: HAVE_IDPROM=GETHOSTID

file: HAVE_ETHER=TRUE

The supplemental version information may change from release to release. Currently, we display which signal handling function is used, how the ID Prom is determined, and if the ethernet address is available or not for host locking.

`jlicver` is handy for determining the release level of the Élan library that a command was compiled with.

6.4.7 elm_resource - Élan Resource File Format

Description

The resource file contains Élan related resource information. The file currently may be used to specify:

- Reserved licenses - licenses that are reserved for individuals, groups, or even machines.
- Held license periods - the minimum period for which a license is held for a user, group, or machine.
- The list of redundant servers, when backup servers are desired.
- An address mask to filter out requests from foreign systems.

Each line of the resource file contains a single definition. A `#` at the beginning of any line will be treated as a comment; the remainder of the line is ignored.

To use a resource file, the license manager daemon `jaleoliced` must be started with the `-r` option, specifying a resource file to use for the license server.

Packet Filtering (Domain Restricting)

By default, the Élan License Management System accepts requests from any IP address. If desired, requests may be limited to only those addresses that conform to your specification. For example, if your company owned a Class-A network, but was also on the Internet, you might wish to limit license server requests from only those on your Class-A network domain.

Packet requests may be limited to machines defined by an IPACCEPT mask specified in the resource file. The mask may appear in the resource file as a line of the form:

```
%IPACCEPT mask
```

where mask is a 4-tuple of expressions e, separated by periods:

```
e.e.e.e
```

Each e may be any one of the following, or a comma separated list of any of the following:

- N: An integer between 0 and 255.
- N-M: An integer range. Each integer should be between 0 and 255, and N should be less than M.
- *: An asterisk matches all integers.
- THIS: The word THIS (or this) means the server's IP octet value in the designated position. For example, if the server address were 192.100.42.16, then THIS.*.*.* would be the same as 192.*.*.* and THIS.THIS.THIS.* would be the same as 192.100.42.*.

For example, a mask of:

```
THIS.*.*.*
```

would limit requests from only those on the same Class-A network as the license server. Also,

```
THIS.100,200-224,226.*.*
```

would limit requests from only those on the same Class-A network and from Class-B subnetworks 100, 200 through 224, and 226.

Server Definitions

Élan may be run with one or more license servers to implement server redundancy (backup servers). In the redundancy model, each server host should be named in the resource file. The server names are listed via lines of the form:

```
%SERVER hostname
```

where hostname is the name of one of the redundant hosts.

The %SERVER definitions are required only for use with redundant servers; they are unnecessary when the single server model is used.

You need keys especially issued for redundant server setups.

Reserved/Excluded/Held Licenses

Feature licenses may be reserved to individual users or client machines via the Élan resource file. Individual users and/or client machines can also be excluded from using specified features. Reserved and excluded licenses are indicated in the resource file via lines of the format:

```
feature:group:client1,...,clientn:k:h
```

The individual items of each line are:

- feature: The feature name, a maximum of 32 characters. (This must be the true feature name, such as "1000", perhaps, not the feature alias ("JaleoComp")).

- **group:** is this group's name, a maximum of 20 characters. Note that this is not at all related to the UNIX system group names in `/etc/group`. It is only the name you wish to call this group of clients. This group name will be displayed during an `jlicadmin -l` style listing.
- **client:** These are either user login names, or host names if preceded by the "@" sign. For example, `jeff` would be a user name, and `@elan` would be a host name. Client names are separated by commas.
- An empty group list means everyone. Useful for setting the hold period only. (Often used with `k` also set to 0.)
- **k:** This is the number of licenses reserved for the group or the word **EXCLUDE** to exclude this group from use of this feature.

EXCLUDE-ed lines should appear before reserved lines in the resource file.

`K` may be 0 to specify a hold time only for this group. An empty group and 0 `k` may be used to set a default hold period for everyone. The hold time is ignored for EXCLUDE-ed lines.

- **h:** The hold period, in seconds, for all members of this group.

When a client terminates, all currently licensed features will be held for this many seconds.

If the same user (actually, same `user@host` and display) runs the application again, these held licenses will be granted to him or her. Thus, held licenses are a form of reserving licenses for active users.

Note that the held period is determined from the resource file as follows:

- (1) If the license requested comes from the reserved license pool for a group, it will have the group's hold period.
- (2) If the license requested comes from the free license pool, the first line that matches the feature and group member will contain the hold time that is used.

Therefore, order in the resource file may be important. (See the example below.)

Example

Here's a sample resource file (using every feature we could think of):

```
# Accept requests from this Class-C network only.
%IPACCEPT THIS.THIS.THIS.*
# We're using 3 redundant servers.
%SERVER hoss %SERVER ben
%SERVER littlejoe
# Our reserved and excluded licenses...
99:mother-in-law:harriet,@wanda:EXCLUDE:0
99:wordproc:wendy,diane,jon,paul:3:600
99:hackers:jeff,sara,tom:3:120
99:lab:@enzo.elan.com,@trager.elan.com:1:0
# Everyone else has a default hold period of 60 seconds
```

```
99:default::0:60
```

In this example resource file,

An IPACCEPT mask specifies that requests should be accepted from the local class-C network only.

Three servers have been defined. Presumably redundant servers are in use. If redundant servers were not used, these lines would be unnecessary.

The user `harriet` or anyone logged onto the machine `wanda`, would be excluded from checking out any licenses for feature “99”.

3 licenses for 99 have been reserved for 4 members of the word processing group, `wordproc`, whose login names are `wendy`, `diane`, `jon`, and `paul`.

3 licenses have been reserved for the members of the group `hackers`.

1 license has been reserved for anyone on the machines `enzo.elan.com` or `trager.elan.com`.

Also, the hold period has been assigned as follows:

If any `wordproc` members run an application requesting feature 99, then the licenses obtained will be held for 600 seconds (10 minutes.)

If any `hackers` members run an application requesting feature 99, the licenses will be held for 120 seconds (2 minutes).

Anyone running on `enzo.elan.com` or `trager.elan.com` will not have held licenses.

Anyone else running the application will have their licenses held for 60 seconds. (Established by the default line.)

Note that if the default group were first in the file, then all licenses that came from the free license pool would have a hold time of 60 seconds. (The way it is, members of `wordproc`, `hackers`, `lab` always get their own hold time, because they came first in the file.)

Note

Order of the lines is relevant:

- (1) EXCLUDED-ed lines should come first;
- (2) If both host names and user names are used, the first match, host or user name, from left to right and top to bottom is used;
- (3) A default hold time should be indicated last.

Thus, in our example above, if `wendy`, while logged into machine `enzo.elan.com`, checked out a license for 99, her license would come from group `wordproc`, rather than `lab`, since that entry appears first in the resource file.

