

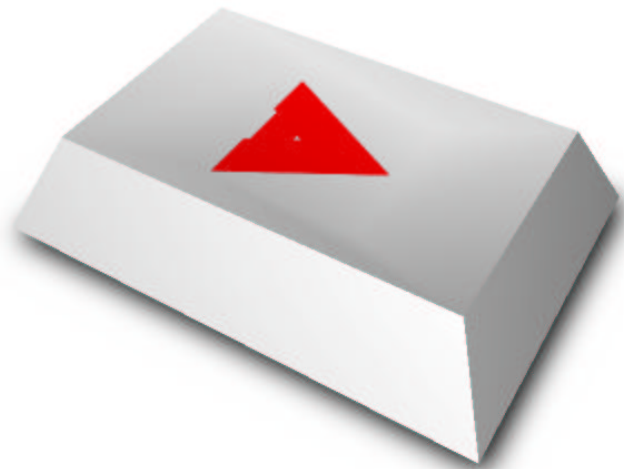
mediAkit and the
NR Open Source Software Center



Norwegian Computing Center/Applied Research and Development

RAPPORT/REPORT

NR medi▶kit



Report No. 996

Wolfgang Leister
Per Røe
Jørn Inge Vestgården
Ole Aamot

Oslo
November 2003

Tittel/Title:
mediAkit and the NR Open Source Software Center

Dato/Date: November
År/Year: 2003
ISBN: 82-539-0503-3
Rapport nr/:
Report no: 996

Forfatter/Author:
Wolfgang Leister, Per Røe,
Jørn Inge Vestgården, Ole Aamot

with contributions by:
Bjørn Nordlund, Eirik Maus

Sammendrag/Abstract:

Open Source software is becoming increasingly important for private end users, governmental institutions, and the software industry. While we recognise that multimedia applications are a vital part of systems today, many multimedia applications are not yet available for the broad public as open source products for various reasons. A group of researchers at NR have taken a closer look at Open Source software (OSS), especially within the field of multimedia applications. The result of this effort is the development of the **mediAkit** distribution, and the NR Open Source Software Center. The goal of our development was to provide an add-on to existing distributions of OSS for Linux, so that the user easily can install and use multimedia applications.

This report also contains overviews on software for building distributions, and a catalogue of multimedia software available for the Linux operating system. The portal pages of **mediAkit** are available on <http://www.oss.nr.no>.

Emneord/Keywords: Multimedia, Distribution, Open Source, Linux

Målgruppe/Target group: NR, research institutions, open source community

Tilgjengelighet/Availability: Open

Prosjektdata/Project data: Channel S, Setup

Prosjektnr/Project no: 11012

Antall sider/No of pages: 80

Contents

I	mediAkit and the NR Open Source Software Center	3
1	Overview	3
1.1	Open Source software	4
1.2	Open Source multimedia projects for Linux	5
1.3	Common Problems with OSS	6
1.4	What is a distribution	7
1.5	What is Open Source	7
1.6	Licensing issues	8
1.7	Open Source and Business	9
2	Developing mediAkit	10
2.1	Developing a Distribution	10
2.2	Package management systems	11
2.3	How to create a software package for mediAkit	12
2.4	Using apt-get and rpm	12
2.5	Licensing	14
2.6	Maintenance of the distribution	14
2.7	Choice of Applications for mediAkit	15
2.8	The mediAkit GUI - specially developed software	15
3	Experiences and Conclusions	17
3.1	Experiences with OSS	17
3.2	Experiences with the distribution	18
3.3	Experiences with NR OSSC	18
3.4	mediAkit and the Future	19
II	The software catalogue	21
A	Tools for building and distribution of software	21
A.1	Build systems	21
A.2	Package Management systems	25
A.3	Install and Update systems	27
A.4	Hardware Detection systems	31
A.5	Bug Tracking systems	33
A.6	Software Image systems	34
A.7	Web portal and content management systems	35
B	Open Source Multimedia Software	37
B.1	Frameworks and multitools	37
B.2	Capturing	37
B.3	Video production and editing	41
B.4	Audio production and editing	42
B.5	Encoders and Transcoders	44
B.6	Player software and presentation	48
B.7	Streaming and other servers	51
B.8	Metadata, search and retrieval	52
B.9	Mastering	52
B.10	Digital TV software	54
B.11	IP phones, multimedia phones and phone conference equipment	55
B.12	Video conference	55
B.13	Text To Speech and Speech to text	56

C Other Multimedia-oriented distributions	57
C.1 AGNULA	57
C.2 DeMuDi: Debian Multimedia Distribution	58
C.3 Planet CCRMA at Home	58
C.4 NIST DASE	58
C.5 MoviX	58
III Details on mediAkit	59
D Details on the DVD Use Case	59
D.1 Creating a VCD	59
D.2 Editing the mpeg-file	59
D.3 Creating a DVD-compatible mpeg-file	60
D.4 Converting a sequence of pictures into a DVD-compatible mpeg-file	60
D.5 Editing and adding sound	60
D.6 Creating a DVD	60
E Software Packaging	62
E.1 Package Layout	62
E.2 Build Instructions	66
E.3 Install Instructions	66
F Software Patches	67
F.1 Making a patch using <i>diff</i>	67
F.2 Applying a patch using <i>patch</i>	68
G List of packages in the mediakit distribution	69
G.1 Programs	69
G.2 Libraries	69
H NR mediAkit on Debian	70
H.1 Packages not yet available in Debian	70
H.2 Packages available in latest stable release of Debian (“woody”)	70
H.3 Packages available in latest testing release of Debian (“sarge”)	70
H.4 Packages available in latest unstable release of Debian (“sid”)	70
I Test reports	71
IV License	73
J Licenses for mediAkit	73
J.1 The End User License Agreement (EULA)	73
J.2 The License Boilerplate	76
References	77
Index	79

Part I

mediAkit and the NR Open Source Software Center

This document is the result of projects at Norsk Regnesentral having the joint goal to learn about open software, and to prepare a distribution of open source software with an emphasis of multimedia. The distribution **mediAkit** is hosted on the NR Open Source Software Center, which is developed to maintain open source software developed at NR.

The work is based on a previous report “Developing a Multimedia Distribution” by W. Leister, E. Maus, B. Nordlund, O. Aamot, and J. Lous [16]. This project is part of the Channel S research program at Norsk Regnesentral in 2003. Other research projects within the Channel S program might also be of interest for the reader. These include work addressing digital TV, image encoding with wavelet technology, and multimedia multi-channel production¹. Especially the project on digital TV [15] is of relevance, since parts of the software catalogues are common to both projects.

In the following we present the principles and the implementation of our multimedia distribution **mediAkit** and the NR Open Source Software Center.

1 Overview

Traditionally the users of Open Source software (OSS) have been experts, early adopters, and developers of other OSS packages. As OSS enters the commercial main stream a new emphasis is being placed on usability and user interface design. Non-experts are attracted to OSS not primarily because of the openness of the code, but rather on the basis of cost, quality, functionality, security maintenance costs, an support. OSS is more than the source code and its functionality: OSS is also the basis for development models, and business models.

While OSS software is available for many areas, we want to address software within the application area of multimedia. Multimedia software is amongst the most challenging software, including advanced algorithms (e.g., codecs), user interfaces, communication between peers, computer graphics, etc. in real time. While many of the components of multimedia systems have been developed on open systems (e.g., codecs), integrated solutions that are working in a plug-and-play manner have been the domain of the commercial actors.²

The development of complex OSS systems like OpenOffice show that the development of stable and complex systems is possible, and creates an alternative to commercial products. The development of complex multimedia applications as OSS is currently progressing, as the development of systems like `xmms`, `xine` or `mplayer` shows, which have advanced functionality (sometimes above the commercial alternatives) and user interfaces (e.g., skins). However, for other areas, like the production of content, streaming, etc. there are not many satisfactory OSS solutions available.

An overall integration of multimedia software to be used in major use cases is still lacking; i.e., the use of this software is still the domain of experts. Another obstacle for overall acceptance of OSS software is that there are many competing projects with similar functionality available, using incompatible interfaces.

The **NR Open Source Software Center** and the development of the **mediAkit** distribution are tools for our research in order to get insight into the development and use of OSS. The mission of the **mediAkit** distribution includes the following goals:

- Chose major use cases for multimedia software; select, package software that has shown practicability for these use cases, and provide these with an easy-to-use installation interface.
- Integrate the chosen application software by developing graphical user interface software that makes the applications easy to use.

¹For more information, see <http://www.nr.no/channelS/projectPlans/projectPlan2002.html>.

²There might be several reasons for this: Developers of special issues often neglect the development of user interfaces, and usability; patenting issues and proprietary solutions for often used services make an adoption difficult;

- The installation procedure must be easy to use, including a modern user interface for installation.
- Use the developer pool of OSS to enable the development of missing parts of the software. Stimulate the participation of OSS programmers by providing the necessary tools, e.g., a repository, web portal, etc.
- Provide the infrastructure to maintain the distribution, including support, bug tracking, mailing-lists, version controlled repository, etc.

The process of designing a distribution is a valuable experience on how to distribute software. Our own experience in other projects show that many smaller enterprises (SME) do not know how to distribute developed software to customers.

1.1 Open Source software

The ideas upon which OSS are founded have existed for several decades. In former days it usually was called “Free Software”. Often, idealistic ideas were behind these thoughts, rather than technical or business reasons. The GNU project [23] (see <http://www.gnu.org>) is a direct successor of this generation, and major components of what many people now call ‘GNU–Linux’, have been developed by the Free Software Foundation (FSF) and its volunteers for the GNU³ system (see <http://www.gnu.org/philosophy/categories.html#TheGNUsystem>) since 1984.

In 1991 Linus Torvalds announced that he was working on a Minix-like operating system kernel. The development methodology of the Linux kernel was different from the GNU project. This is described in the article “The Cathedral and the Bazaar” by Eric S. Raymond [22], where this type of software development is summarised in the slogan: “Release early, release often”.

After the Linux kernel became very robust, more developers adapted to the Linux development model and the way of distributing software, and the Linux project became a model for open source development. Other large projects like Mozilla, StarOffice / OpenOffice.org followed.

Open source as a business model was released about the year 1999, with Eric S. Raymond as a spokesman [22, 6]. Several program packages have been released using the open source model, and open source products are more often used in business, public administration, and schools (see, e.g., the Skolelinux project [1]).

The phenomenon Open Source is discussed in several seminars and publications, which still form the reference literature, e.g., the book “Open Sources” [6]. Besides the definitions and development models of open source, also business strategy [3] is an important topic.

The main thought behind open source products is not necessarily, that these are at no costs. While most of the software comes with a license that permits use of the software for free, installation, maintenance, support, and the use of other services usually causes expenses. It would be dangerous for enterprises to neglect these costs, even if these costs are to pay own employees for their time.

The openness of the code in OSS gives several advantages:

- The user can inspect the code and consider its quality.
- The user can see whether problems could occur, e.g., with regard to security issues.
- The user can make changes, adjust the product for his use, and fix bugs that might occur.

The development model of many open source projects is quite different from commercially driven development projects: In OSS projects the project owners are responsible for coordination of many volunteers, maintain a database, take decisions, perform some of the programming, etc. Users of open source applications can suggest changes in the software, and thus contribute to further development. For most of the projects web pages are available, including possibilities for download, information on the project, communication between developers, addresses for reporting bugs, etc. There are also communities that house many projects, e.g. Sourceforge (<http://www.sourceforge.net>) and Savannah (<http://savannah.gnu.org/>). The distribution model for OSS is often based on web servers (or ftp servers) with download facilities. Many projects also have mirrors in order to avoid overload of the download servers.

³GNU refers to a recursive acronym for “GNU’s Not Unix!”.

The term **distribution** is used for software that is bundled together. Over 300 Linux distributions with emphasis on different subjects are available (see <http://www.distrowatch.com>). Distribution projects with an emphasis on multimedia are presented in Section C of Part II.

OSS is also distributed by several organisations within several application areas. Most of these offer repositories on web sites using search engines, portals, and other mechanisms for maintenance of software. Examples include [Mathtools.net](http://www.mathtools.net/) (<http://www.mathtools.net/>), Linux Online (<http://www.linux.org/>), The GNU Free Software Directory (<http://www.gnu.org/directory/>), Sourceforge (<http://www.sourceforge.net>), and several others.

Soon after the CDROM was detected as distribution medium the “Prime Time Software for Unix” [18] and “PowerTools for Unix” [19] were released as distributions of software.⁴⁵ These can be seen as early attempts to create a distribution of useful software for the user. A book or booklet, and installation programs followed with the CDROMs, in order to make it easier for the user to install the programs.

Other interesting areas for putting together a distribution include: E-learning, E-government, subjects from sciences (biology, physics, mathematics, etc.) or medicine (see [CDmedic http://cdmedicpacsweb.sourceforge.net](http://cdmedicpacsweb.sourceforge.net)).

1.2 Open Source multimedia projects for Linux

The Linux operating system is well suited for multimedia software. Several embedded architectures (e.g., for digital TV or mobility applications) use Linux as operating system. Since software packages are from different origin using many programming paradigms setup procedures often are confusing for the common end-user using a PC. Many packages require special skills to set up. The lack of integration, and dependencies on special resources (such as dynamic libraries) which the end user must search for and install manually often leave unsatisfied end users, who rather would prefer an off-the-shelf product.

OSS within the area of multimedia often lacks functionality and integration with other products. Most of the solutions are not yet mature for the mass market. In order to become main stream much must be done within a relatively short time, since otherwise users will be driven away from OSS.

Examples for multimedia distributions using Linux include the AGNULA project, the DeMuDi, or Planet CCRMA at Home (see also Section C). An interesting development is the deployment of distributions that do not need to be installed, but which run directly from a CD. These distributions are used for demonstration purposes (e.g., Knoppix), and in order to distribute player software and operating system together with the media files (e.g., eMoviX).

Self-contained distributions that run directly from the installation-medium have one advantage: the software does not need to be installed. However, some limitations to the access of resources (e.g., memory and swap files) and the fact that multimedia files take much resources in memory made us postpone such plans for our own distribution.

While many distributions contain everything necessary of the operating system, applications and documentation, the first version **mediAkit** is meant to be an easy-to-install **add-on** to the ordinary distributions; i.e., we only focus the software relevant to our application cases, and let others handle the distribution of the basic software. Our distribution **mediAkit** will be a supplement to an existing distribution, like like RedHat, SuSE, Debian, or Knoppix.

The Gentoo distribution (<http://www.gentoo.org>) distributes all program packages in source code. Besides avoiding some juridical problems, source code distributions allow to generate optimised and specialised versions for a target machine. However, due to lengthy compilation processes during installation, and the fact that package managers like RPM, and *apt* do not support source code dependencies between packages, we decided to use binary installation packages instead.

A catalogue of multimedia software can be found in Chapter B of this report. This includes to take decisions which packages to choose, and to classify and prepare the packages. In order

⁴In former Red Hat releases there has also been a Powertools CD as an add-on, which is different from the described Powertools for Unix.

⁵The Powertools for Unix is released in a new version recently.

to find relevant projects search engines at relevant sites (e.g., <http://www.linux.org/apps/>) or sites that host projects (e.g., sourceforge) have been consulted.

It is a general observation that there are many open source multimedia projects available, but many of these are not (yet) included in “standard” Linux distributions. In our current project, we have gathered an extensive list of what seems to be the most relevant and useful open source multimedia software available.

For many of these projects the status is unclear; some projects are active, while others are abandoned, are no longer supported, or have died. The quality of many projects can be inferior than what can be expected from commercial products, especially with regard to documentation.

An extensive list of possible software packages is listed in Part II of this report. We are aware that this list only can be a selection of available software within multimedia. On the other side, not all of these are included in the **mediAkit** distribution. In some cases several packages with similar functionality are included into **mediAkit**.

1.3 Common Problems with OSS

Today the availability of multimedia software is vital for an operating system to succeed in the market. For several reasons, multimedia software is not always available for Linux. Not all decoding algorithms for multimedia are available as open source⁶, while they are available on commercial alternatives. Many potential users are thus chased away from using open source products as a consequence of these policies. The unavailability of some kinds of multimedia software makes that users no longer use Linux as “their” operating system.

To show an example, we look at the possibility to play DVDs on Linux. While the technology is available, there are laws that forbid the distribution of the unscrambling system for DVDs (called CSS), especially in the United States. For Europe the laws are more unclear on whether it is possible to distribute the DeCSS software. The DVD players distributed in major distributions often play only unscrambled DVDs, while the common user does not understand why her legally bought DVDs cannot be played on open source operating systems.

A commercial DVD player for Linux has been announced for a long time: LinDVD by Inter-Video with a price tag of about 30 US\$. However, a look at the web pages of the provider shows that LinDVD is only available for manufacturers for evaluation and integration. Therefore, in reality there is still no legal (seen by the eyes of the DVD industry) possibility to play all DVDs on Linux for the normal end user.

Other obstacles for the introduction of multimedia software in Linux includes patent issues and closed media formats, where reverse engineering is prohibited. Sometimes Win32-libraries are altered and used, but also this possibility can cause copyright problems. This is, however, more an issue of open standards than OSS. However, Open Standards and OSS are also closely related to each other [14].

There are also technical obstacles why it is hard to distribute multimedia software. Often the processing power necessary to present multimedia content is rather high, and special optimisations are required with respect to hardware (e.g., CPU, cards, periphery). These requirements often need a more profound knowledge than the common user has. Auto-detection techniques with respect to software configuration is often not developed to the necessary extent.

Another example includes the production of video on DVD. There are several standards (+R, +RW, -R, -RW) that must be supported, while the standards only partially are openly available. The software available for DVD burning includes open source software (like *dvdrecord*), and closed source software (like *cdrecord-ProDVD*⁷).

Tests with older drives using the -R and -RW standards showed that *cdrecord-ProDVD* worked, while *dvdrecord-1.3* failed⁸, and *dvdrecord-1.4* worked. Experiments with a NEC ND1300a (which is able to write all of the DVD standards) showed that *dvdrecord* failed, while *cdrecord-ProDVD*

⁶Many of the multimedia data formats are patented or closed, in order to prohibit knowledge of the software. Therefore, decoding software cannot be developed legally as open source.

⁷The software is a closed source development based on *cdrecord*, by the original author of *cdrecord*. The program can be used for personal purposes free of charge. Due to licensing reasons **mediAkit** cannot include *cdrecord-ProDVD*.

⁸*dvdrecord 1.3* is included in RedHat 8.0.

could only write DVD-RW; DVD-R had trouble with random writing errors, DVD+R(W) could not be written⁹ at all.

In the Linux 2.4 kernel there is no support for writing DVD+RW disks directly; while the UDF file system is marked as experimental. However, there is a patch¹⁰ which allows the writing operation to DVD+RW. Since this patch usually is not included its installation would require the generation of a new kernel each time; therefore this patch is impractical for our purposes. The new Linux 2.6 kernel will contain the possibility to write to DVD+RW disks directly.

As a conclusion, these experiences reveal that open source products still are not mature enough to be used in major applications. It is also a general observation that open source products often are one step behind for some popular technologies (while open source products might be ahead for other technologies, which not yet have reached market maturity).

1.4 What is a distribution

According to the dictionaries the term **distribution** refers to the commercial activity of transporting and selling goods from a producer to a consumer.¹¹ The Jargon File [25] (version 4.2.3, 23 Nov 2000) gives the following definition:

1. A software source tree packaged for distribution; but see *kit*. Since about 1996 unqualified use of this term often implies 'Linux distribution'. The short for **distro** is often used for this sense.

The term *kit* refers to the following:

[Usenet; poss. fr. DEC slang for a full software distribution, as opposed to a patch or upgrade] A source software distribution that has been packaged in such a way that it can (theoretically) be unpacked and installed according to a series of steps using only standard Unix tools, and entirely documented by some reasonable chain of references from the top-level README file. The more general term *distribution* may imply that special tools or more stringent conditions on the host environment are required.

The definition of the word *kit* already contains some hints on how an early distribution of software was organised, and what it contained. The expectations of the user are higher today with respect to automatic setup routines, and therefore a distribution in that manner would no longer suffice.

1.5 What is Open Source

The definition of Open Source is varying. However, there seems to be a common understanding on the definition, when we set aside some attempts of commercial actors to undermine the Open Source movement. According to the Open Source initiative at www.opensource.org the distribution terms of Open Source software must comply with the following criteria:¹²

1. **Free Redistribution.** The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
2. **Source Code.** The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicised means of obtaining the source code for no more than a reasonable reproduction cost; preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

⁹We got an error message that the image is too big, even though the total image was no bigger than 1GB.

¹⁰The DVD+RW tools are available at <http://fy.chalmers.se/~appro/linux/DVD+RW/>.

¹¹There are other meanings of the word **distribution** which are not relevant for our purposes.

¹²The rationales on these terms can be found at www.opensource.org/docs/definition.php or [20].

3. **Derived Works.** The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
4. **Integrity of The Author's Source Code.** The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
5. **No Discrimination Against Persons or Groups.** The license must not discriminate against any person or group of persons.
6. **No Discrimination Against Fields of Endeavour.** The license must not restrict anyone from making use of the program in a specific field of endeavour. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
7. **Distribution of License.** The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license (e.g., non-disclosure agreements) by those parties.
8. **License Must Not Be Specific to a Product.** The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.
9. **The License Must Not Restrict Other Software.** The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.
10. **The License must be technology-neutral.** No provision of the license may be predicated on any individual technology or style of interface.

An overview, taxonomy, and analysis of Open Source Software, and its development process can be found in [10]. In the taxonomy OSS is compared to products that are commercial, trial software, use-restricted, shareware, freeware, royalty-free binaries, royalty-free libraries.

1.6 Licensing issues

In order to maintain the openness of the software, licensing is an important issue. There are many licenses that are compliant with the criteria of Open Source. An overview of licenses including comments can be found at GNU.org [12]. There are many other sources with different viewpoints, partially dependent on the business strategy. See, e.g., [3] or [20].

Software needs licensing in order to define the terms for the distribution and usage of the software. The copyright holder must define the terms for the distribution. Many licenses can be applied, both commercial and open source. However, the definition of Open Source restricts which licenses can be considered as an Open Source license. While there are many licenses, we can categorise these into several classes:

- **Freeware and Public Domain:** The copyright holder gives the program away, without keeping any rights.
- **BSD-style:** The BSD-style¹³ licenses permits the program to be copied, modified and incorporated into both open- and closed-source software. The license requires that credits to the authors of code within the source code and documentation are intact. The license permits that the program code can be distributed in closed form, not requiring that improvements are coming back to the developer community.
- **GPL/LGPL:** The GPL (GNU Public License) is designed to keep open software open (free). However, there are some restrictions how to use GPLed software, which makes it sometimes impossible to mix proprietary with GPLed software, without implications that are not acceptable for many businesses.

¹³BSD stands for Berkeley Software Distribution, see www.bsd.org.

The GPL has some possible drawbacks for business, which often is caused by the “viral” behaviour of the GPL: Software under the GPL cannot be combined with non-GPL software without this software also being under the GPL. When contributions from open source programmers to a project are included into a project, the changes cannot be included into software following other licenses without consent of the contributor.

- **Mozilla-style:** Mozilla License (Netscape) is a special license for the business situation Netscape was/is in, and should therefore not be used for other packages, when these reasons are not there (contracts with others).

This is one of the reasons for the “Mozilla” license, where contributions should be used for non-open source releases of this software. Mozilla.org is now relicensing the whole packet with a triple license, which is a tedious work.

The copyright holder can license software using several licenses at the same time (for the parts the copyright holder has the rights for)! This practice is used in the case when there is a commercial branch of the software, or there might be patent problems or incompatibilities with parts of the software licensed by another license (e.g., Mozilla. see <http://www.mozilla.org>).

For a distribution we need:

- The EULA (End User License Agreement). This document sets the licenses for the distribution, and its components. Due to the fact that a distribution consists of many sub-components, each with a separate license, this document points to these licenses. In addition this document contains disclaimers.
- A set of guidelines or the social contract of the distribution, i.e., guidelines on how to contribute, on which terms, and what software is included. Example: The Debian Social Contract [21], which went into the definition of Open Source (c.f. www.opensource.org and [20]). The definition of Open source [20] must be followed.
- The licenses for the setup software specific for the distribution. The setup and maintenance software can be considered as a software package, and goes therefore under the next item.
- The licenses for each software package included. Each software can have a license of its own. However, the license must be compatible with the guidelines of the distribution.

1.7 Open Source and Business

The most important issue of open source products within business includes which business model to use in order to earn money, which is not very obvious. While some models have not been a success the following four models are promising [17], which have their roots in business models for SME:

- Use of dual licensing, suitable for development tools that can be used in end user programs that must be payed for.
- Support of open source software is the main business.
- For computer games the software is open source, while data which make the game interesting must be payed for.
- Main business is something else, e.g., hardware components.

However, there are other settings for open source, e.g., the public or interest organisations financing the development of open source products. Also large companies seem to fund the development of OSS in order to follow a long term strategy. OSS can also be used in order to open new market places.

2 Developing mediAkit

This section covers decisions and solutions made in the current project, including choice of software to be distributed, technical decisions on routines and maintenance of the distribution. Since the former report of this project [16] was written, we had to update and revise some of the decisions taken, since they showed to be impractical.

A distribution contains several software packages that can be installed on a computer, using a unified installation and maintenance procedure for all software packages. Besides the software, the infrastructure part of a distribution includes installation procedures, a software base, query scripts, software configuration tools, package management, etc. Additionally a bug tracking data base, tools for cooperative work, and a Web server for storing the distribution are parts of the total project content. Maintaining the distribution we also must be prepared to receive contributions and providing new versions.

In **mediAkit** we emphasise software to produce, store, serve, stream and play multimedia (sound, images, video, etc.).¹⁴ **mediAkit** is built on use cases, where software to these use cases has been collected, tried out, and packaged. We have focused on software for capturing multimedia data (using microphone and camera), editing, playback, encoding and distribution. We have gathered an extensive list of relevant software projects (see the software catalogue in Section II).

The first version of **mediAkit** is now available for Red Hat 8.0 as the first basis distribution. Other add-on distributions of **mediAkit**, e.g. for Debian, are under preparation, and are intended to be developed with the efforts of the open source community.

We plan to create a distribution on a single bootable stand-alone CDROM, or pendrive. This might be implemented as a derivative of the Knoppix (see Section C.5) distribution without e.g., text processors and other software not relevant for multimedia.

2.1 Developing a Distribution

For each software package we have three distinct tasks: (1) collecting, configuring and packaging the software package, (2) installing the software package at the end user site, and (3) maintenance of the distribution. All of these are supported by specialised tools.

In the **packaging task** the following steps are performed:

- configure software packages, find dependencies, retrieve dependent software, compile and test the software, etc.;
- the software packages are packed together using a package management system (PMS);
- software for installing the distributed software packages at the client side, and to perform queries must be added to the distribution;
- the tools necessary for doing installations in the particular basis-installation must be prepared.

In the **installation task** the following steps must be supported on the end-user computer:

- Queries whether the software already is installed, and in which version;
- unpacking the software, configuring, and installation.

During **maintenance** we emphasise on the following:

- **mediAkit** provides routines for preparing software packages that are newly added to the distribution. Deploy routines on how to include new versions of already included software;
- a build-system for distribution and installation must be employed which handles interdependencies, hardware- and software dependencies;
- bug handling system, forward bug reports, or suggestions for problem solutions to those responsible, discussion groups on distribution-related issues;
- provision of a CVS base on a server;

¹⁴Note, the principles and routines developed for **mediAkit** can also be employed for distributions with other content.

- maintenance of distribution: Check whether new versions, bug fixes, etc. of original software are available;
- automatic build system regularly in order to detect problems immediately;

Each software package consists of a set of files that contain the executable code, library functions, setup information, etc. Multimedia software tends to have a more complex setup than ordinary programs, because of possible plugins, codecs, and other dependencies, which must fit together. All files are generated from a source code base. In order to generate the executable programs tools are used, e.g., software configuration tools, compilers, linkers, etc.

The setup procedures of the distribution must support the generation of the necessary files from the distributed file of the software package. All these files need to be suitable for the appropriate version of the software to be installed. When installing a software package, the administration software of the distribution must unpack the file, detect dependencies between files, possibly require other packages, compile, configure and install the package.

Hardware dependencies, version numbers of software, and especially dynamic libraries, introduces additional complexity to the installation procedure. In order to reduce this complexity, different systems for managing packages are available, see Sections A.2 and 2.2.

While working with the project we encountered another source of complexity: Often software packages, especially libraries, must be available in several versions in the same distribution, since the software package might have branches with different functionality.¹⁵

All steps in the distribution process are heavily supported by programming tools, like *make* or *autotools*, in order to provide more automated generation of software packages. See Section A.1.2 for further details.

2.2 Package management systems

A package management systems (PMS) is used to pack and distribute software, mainly pre-built libraries and programs in binary form. Reasons for distributing software packages in compiled form (vs. distributing the source code) include:

- Configuring, compiling and linking is time-consuming and prone to errors. Installation can be done much faster if the modules are already pre-built.
- Most users only want to use programs, not alter them. Thus, they do not want or need the source code. This is especially true for desktop users of Linux. Some newer Linux desktop installations do not even install development tools such as compilers per default (unless the user selects “install all”), since many users don’t intend to do any programming anyway.

However, for technical and juridical reasons, we considered to distribute the software entirely in source code.¹⁶ There are distributions of Linux that use a source code distribution (e.g., the Gentoo Linux Distribution (<http://www.gentoo.org>), but experience shows that installation time is rather high. Additionally, the common PMS do not support source code distribution and dependencies in source code sufficiently.

PMS are tool suites that ease distribution of software packages, managing dependencies and version control. These are normally built around a database that keeps track of installed modules, provides version information, and resolves dependencies between the various modules. A PMS also provides means to query a database, install new modules, and uninstalling modules. An overview of PMS is given in Section A.2.

An overview of binary package formats used for Linux platforms can be found at <http://www.kitenet.net/~joey/pkg-comp/>, where the formats *deb* (used by Debian), *rpm* (developed by Red Hat), *tgz* (used by Slackware), *s1p* (Stampede Linux), and *pkg* (SVr4 package format for Solaris) are compared. Debian and the Norwegian effort “Skolelinux” [1] (see <http://www.skolelinux.no>) is built on top of the **Dpkg** system developed by Debian (see section A.2.1), while other major

¹⁵We encountered this problem with the *mpeg2dec* program, see Section B.5.12. While a necessary patch to an older version is available, we also need functionality of the newer version, which is linked to an incompatible library; therefore at least two versions of both libraries and programs must be installed.

¹⁶The juridical reasons include the fact that some software, e.g. software for encoding MP3 streams, cannot be distributed as binaries. The technical reasons include possible optimisations of code, which could be vital for multimedia software.

Linux distributions, including Red Hat, Mandrake and SuSE, are based on the Red Hat Package Management System RPM [2] (see section A.2.2). Between some of the distribution package formats converters exist, e.g., Alien (see www.kitenet.net) which converts between deb and rpm formats.

RPM does not provide configuration, building and installation from source code packages using a *single command*. While RPM supports the distribution of source packages it does not support resolution of dependencies between source packages. Therefore, RPM is not suited for distributions based solely on source code, since RPM only supports the definition of requirements for installed binary packages.

PMS are usually not inter-operable, nor do they share the same data base on installed modules. The distribution format of the PMS is based on different file formats. Each installation should have only one PMS in use at the same time. Note, that due to different file naming standards binary packages using the same PMS, but from another distribution might and might not work.

Some experiences with the packaging system

We did not find a packaging system that could handle both binary packages and source code packages well. We looked especially at *apt* and *rpm*. However, the architecture of both systems seem to be optimised for the binary distribution model.

The problem with binary distributions (apart from some packages having licenses that prohibit binary distribution), is connected to the use of shared libraries. Almost all application binaries are linked against some shared libraries. In some cases the version of one library is not binary compatible with another. Therefore, an application linked against version *x* of a library cannot be linked against version *y* without recompiling the application.

To avoid difficulties the machine where the package is installed must have the same versions of the libraries as the machine where the package is created. This can often result in packages that work on only one Linux-distribution, or sometimes work with only one specific version of the Linux distribution.

We encountered such a problem when including the *audacity* package, which links to the *wxGTK* library. The first version of audacity in our distribution linked against version 2.2 of *wxGTK*. During testing the site *freshmeat*, where we get our standard packets from, upgraded to version 2.4 of *wxGTK*. This resulted in that the wrong version of *wxGTK* was installed, and therefore *audacity* did not work. To fix this we had to create a version of *audacity* that links to the 2.4 version of *wxGTK*.

2.3 How to create a software package for mediAkit

mediAkit in its first prototype is based on Red Hat 8.0, which uses the RPM package management system (see Section A.2.2).

For the process of creating a software package for **mediAkit** it is a preliminary that the software package is available as source code for the Linux platform, and is tested by the implementation team. Then the following steps are necessary (See Section E for a packaging example):

- Prepare the software packages to meet the requirements (e.g., use of autotools, make mechanism, etc.).
- Place the software at the correct place in the source tree of the distribution.
- Create a spec file (see Section E.1.3).
- Create a Source RPM package with *rpmbuild* (see Section E.2).
- Build machine specific Binary RPM packages from the Source RPM package with *rpmbuild*.

2.4 Using apt-get and rpm

This section gives a step for step introduction to how apt-get is used for an rpm-based repository. Apt-get was designed for the Debian package system, and was modified by Conectiva for use with rpm-packages.

2.4.1 RPM

RPM, the Redhat Package Manager, is a system for installing and maintaining software. A rpm-package usually contains the program-files for installing the software, together with a description of where to install the different files. RPM also maintains a database over installed files, which makes upgrades and removal of packages easy.

In addition to the binary rpm-packages, source code packages (so-called source-rpm) can be created. When these are installed the source code installed, from which binary rpm's can be built using the *rpmbuild* command. The resulting rpm can be installed in the ordinary way.

RPM checks for dependencies: When a new software package is installed, rpm queries the local rpm database and checks that all necessary packages already are installed. If a necessary package should be missing, rpm will stop the installation with message that tells about which package(s) are missing.

2.4.2 apt-get

Running rpm manually can be quite tedious, since frequently packages are missing which must be downloaded and installed prior to the package to be installed. Apt-get solves this problem by installing automatically all dependent software packages. Apt-get can also query for installed packages, upgrade packages, upgrade the entire distribution, or remove packages. A closer overview on the use and functionality of *apt-get* can be found in a recent contribution to the Linux Journal [13].

How to make a http based apt repository. Apt uses a software repository with a certain structure and generated in a specific manner. Usually the programs *genbasedir* and *genpkglist* are used for this tasks. A software repository consists of the directory structure containing `./[architecture]/RPMS.[name]` and `./[architecture]/SRPMS.[name]`. In the **mediAkit** repository the `[architecture]` part is `/redhat/8.0/en/i386` while `[name]` is `mediakit`. All binary rpm files are put in `SRPMS.[name]`, while source rpm files are put into `SRPMS.[name]`. To generate the the apt database(s) type the following:

```
genbasedir --progress --flat --bz2only /.../[architecture]
```

which will hopefully make the directory `./[architecture]/base` and fill it with apt database files. Note: If you see a message on signatures missing the apt-database will not be correctly generated. To import the signature¹⁷ type

```
rpm --import signature-file
```

After the apt-repository is correctly generated the entire directory structure must be moved to the web server.

Example of using a rpm-based apt repository. An apt repository consists of a directory with the architecture name, and a directory containing source packages. Under the architecture directory a directory exists containing the binary rpm files and a base directory, which contains a list of all packages in the repository, *pkclist.[dist]*, and their dependencies.

The base directory also contains *srclist.[dist]* which contains the contents of the source package directory, and their dependencies. Be aware that these dependencies link to binary packages, not source packages. Note that the names of binary packages and the corresponding source packages can differ!

Before using apt-get to install packages the contents in the repository must be updated by running **apt-get update**. This command updates the local package-list. Apt can handle several repositories at the same time, independent from the server the package originally was installed from.

¹⁷The file `/usr/share/doc/apt-0.5.4cnc9/RPM-GPG-KEY` is used in our installation for Redhat 8.0 as the signature file.

To install a package the command `apt-get install package_name` must be run, which lets search the local package-list for the requested package. Then it checks whether all the dependencies are installed, and generates a list over all packages that must be installed in order to install the requested package. It then downloads all these packages, and installs them.

2.5 Licensing

All of the packages of **mediAkit** have separate licenses, which remain untouched when being part of a distribution. However, we added an End User License Agreement (EULA) to the distribution which makes this fact clear (see Part IV). Besides the open source software, **mediAkit** can also contain licensed software, for which a separate, non-OSS license is applicable.

For the installation software of **mediAkit** we ended up with a model similar to `mozilla.org`, who use a triple license MPL/LPGL/GPL. We follow the same line: The setup software for **mediAkit** is triple-licensed: NRPL/GPL/LPGL. As a consequence all contributions to **mediAkit** must follow this licensing scheme in order to be accepted by the development team. The NRPL permits NR to use the software within own projects without limitations, herein also the use in closed source projects. Following this license NR is allowed to use changes and contributions from third parties in order to produce a distribution of proprietary software to NR's customers. Due to the triple licensing all changes made by third parties still follow the rules for Open Source Software.

In the discussion whether other licensing schemes are useful for **mediAkit** the following considerations were taken:

- A license only using GPL/LPGL would not permit NR to use the **mediAkit** code with changes in proprietary distributions. For the **mediAkit** distribution we are interested in being able to use parts of the software for other distributions, possibly non-open source¹⁸. Despite the explicit wishes from GPL advocates, a triple license is suitable for our purposes, since the sole application of the GPL would restrict our business too much.
- Licenses of the BSD-type does not give incentives for external developers to contribute changes back to the open source community.

2.6 Maintenance of the distribution

New versions of software are developed continuously, including new features, corrected bugs, etc. Therefore, efforts must be made to keep the distribution up to date. For maintenance, automated routines must be employed to follow up new developments, update existing software packages, etc. Additionally, an infrastructure for reporting and forwarding problem reports, and a exchange of messages, etc. is necessary.

The **NR Open Source Software Center** web server has two purposes: offer the distribution and its files, and offer a forum for problem reports, support, etc. We first used Postnuke (See Section A.7) as a portal. However, this solution was not taylored enough for development of OSS. Therefore, we installed *gforge*, which is a branch from the sourceforge software. The portal for the **NR Open Source Software Center** is accessible at <http://www.oss.nr.no/>.

For the automatic update of software, we implemented scripts to check the availability of new versions of a software package. However, this makes use of conventions that are not used by all implementors. Normally, a general link to a software package points to the newest version, and when the md5 signature¹⁹ changes, a new version has been published, which triggers the regeneration of a new version of the software package in the distribution.

The setup-script is in the database of *gforge*. On the machine `oss.nr.no` the files for **mediAkit** are located in the subdirectory `/var/www/gforge/www`, which contains the repository for the distribution in the subdirectory `redhat/`. Files should not be added manually to the repository, since there is a mirror of the repository located on the same machine under `/setup/`.

¹⁸As a research institutions for applied research we have to acknowledge the wishes for our customers, not all of which want to listen to the argumentations for open source.

¹⁹We maintain a database with md5 signatures, and the script checks regularly whether this signature has changed.

To add or modify a package in the distribution the following method is used:

1. Make/obtain the binary rpm, and possibly the source rpm of the package.
2. Copy the binary rpm to `/setup/setup_cd/htdocs/pub/mediakit/redhat/8.0/en/i386/RPMS.mediakit`
3. Copy the source rpm to `/setup/setup_cd/htdocs/pub/mediakit/redhat/8.0/en/SRPMS.mediakit` and set the correct permissions to the files.
4. Run the following shell script: `sh /setup/MakeRepository.sh`.

This will compile the base-directory for the distribution, copy all the files to the web-server, and make sure that the permissions of the files stay correct. Additionally a CD-image is created, and placed under `/setup/iso/`, which is copied to `htdocs/pub/mediakit/iso/` on the webroot directory.

New releases of the file `setup.sh` or the iso image of the distribution must be inserted by uploading in the gforge function “files” for the **mediAkit** project.

2.7 Choice of Applications for mediAkit

While we can think of many scenarios, we chose some generic and common cases. For these cases we provide packages that contain all dependent resources necessary to download the software automatically. These packages also include an easy to use user interface software for these tasks, in order to avoid that the user needs to specify complicated command line instructions.

Inspired by a recent issue of LinuxMagazin on producing DVD from home videos [7], we developed the first case: A user produces a DVD/VCD from self-produced video and other available multimedia material, including the possibility to edit sound, video, and still images.

The use case of the application includes the following tasks:

1. import data from source (e.g., video camera);
2. edit video, cut, add sound, add still images, add other multimedia content, etc.;
3. format conversions between different formats;
4. mastering DVD, or (S)VCD, and produce the medium;
5. play content from medium and from files.

We identified packages for the different tasks, and included these into **mediAkit**. The use of the software was rather complex using the command line interface. See Part III for how to use the software packages in this manner, which is far too complex for the average user of a desktop computer. Therefore, we developed two simple user interface programs, which are operated graphically (see Section 2.8).

2.8 The mediAkit GUI - specially developed software

Easy-to-use interfaces are vital in order to integrate the software packages of **mediAkit**. We developed two GUI applications: **mediAkit GUI**, which is a small GUI to start tasks in **mediAkit**, and **mediAkit DVD**, which is a GUI for burning DVDs and VCDs. Both programs are developed within the same CVS tree: `setup-GUI`.

mediAkit GUI is used to call other GUI applications, including *Kino*, *MPlayer* and *Mediakit DVD*.

The **mediAkit DVD** lets the user create a list of mpeg-files, and the application burns these files to a DVD or VCD when the burn button is pushed. When burning DVDs and VCDs several console applications are called. For flexibility, this part is handled in shell scripts, which, when installed, are put at `/usr/share/mediakit`.

GUI Development with Glade

Both GUI applications are developed with Glade, GTK and GNOME. *Glade* is used to design the GUI, and to define the behaviour of graphical objects. The output from *glade* is an XML-file with the extension `.glade`. *Glade* can be used in two ways:

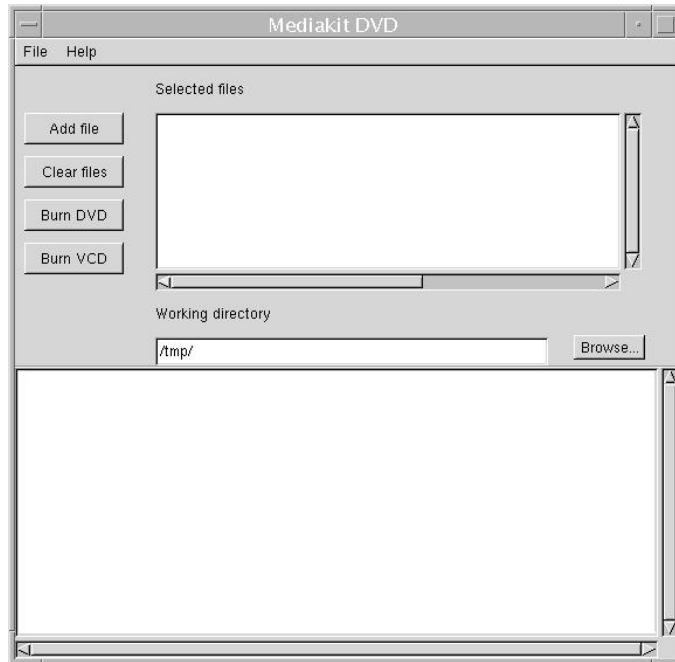


Figure 1: Snapshot of user interface for burning a DVD

- *glade* generates code, i.e., generates a file with callbacks where the programmer adds code. C is the default language, while several other languages are supported with additional installations.
- The approach that we chose is to use *libglade* to load the *.glade* file dynamically after the application starts. The benefit with this approach is that there is no interference between the automatically generated XML-code and the application code. After the application starts the XML-file is loaded and the corresponding GTK-objects are created.

In both cases the connection between the GUI and the application is done by signal handlers in the application, which are called when window events occur.

We encountered several fundamental problems with the development tools during development:

- Lack of documentation. Especially for *glade* and *libglade* the available documentation was rather poor, and sometimes even misleading. (GTK and Gnome are quite well documented, though.)
- Version confusion. GTK has come with a new API in version 2, with corresponding Gnome2 API. In the documentation it is recommended to use the version 2 APIs. However, the poor condition of *Glade2* made this currently impossible.
- Gnome is an extension to GTK, and the Gnome API contains some useful functions that the program makes use of. Unfortunately the semantics of the Gnome objects does not allow the application to fork, resulting in only GTK objects being used in the forking part of the application. The reason for the problem is that the Gnome-widget event-queue gets confused by the fork and responds with an infinite recursion. (The GTK event-queue behaves sanely in the same situation.)

3 Experiences and Conclusions

We developed the distribution **mediAkit** which contains several multimedia applications for use in Red Hat 8.0 base distributions. The use case of **mediAkit** in the current release is to produce DVD from own camera recordings. The **mediAkit** distribution consists of the applications, an installer, a gforge installation, and other routines for maintenance of the distribution.

3.1 Experiences with OSS

Relations to the Channel S program. The goals of the Channel S program include to “provide technological building blocks which can serve as a basis for commercial products or services related to service channelling”. The availability and use of open source products within service channelling is an important issue within Channel S.

Based on the use cases, the software catalogue, and our experiences in this project we conclude the following:

- OSS is available for many major use cases, in some cases with features beyond commercial products. However, much of the OSS designed for the end user lacks of integration, suitable user interfaces, and often major functionality.
- The use of closed standards by commercial actors often make multimedia applications on open source platforms look clumsy and behind the edge of technology. However, interesting open source products are often adapted by commercial actors with minor efforts than the other way around.
- Most multimedia related applications are still in a development state, with major functionality missing. However, in some areas state of the art products are available (e.g., *gimp* for image editing).
- Some technologies are still difficult to use with Linux. While the use of firewire went fairly well, the process of writing DVDs is still a problem, and needs improvement.

Experiences with OSS applications. Many OSS applications are available for multimedia. However, the support is varying for different areas of use. While application areas like programming environments or office applications are quite good integrated lately, multimedia applications are not as good supported and integrated. The reason for that might be the complexity of multimedia software, and its hardware dependencies. Beyond that we have the following observations:

- Applications for viewing and playing multimedia are available, and often easy to install and easy to use. However, in many cases the applications are not integrated well enough for an end user. For instance, applications for viewing DVD is still lacking menu-support and viewing possibilities for additional materials (which is a functionality which can even be found on free-of-charge applications delivered when purchasing a DVD drive for Windows-family operating systems).²⁰
- In several occasions we observed compatibility problems and bugs in OSS, which often is at a beta state.
- Importing multimedia content from external devices (e.g., camera) are available and work satisfactory. However, the user interfaces are very often at a rudimentary level. When a graphical user interface is available, these are mostly too simple for use in real-world applications.
- Software for producing content beyond editing files with a text editor is rarely available. For editing video material several OSS applications exist. However we only found the application *kino* (See Section B.3.2) promising, despite of some obvious lacks of functionality.
- For audio editing several tools exist, and the developer base seems to be larger than programmers developing video editing applications.

²⁰There exist several programs for playing DVDs on Linux with menu-support. Ogle has had menu-support for several years now, and Xine has had menu-support as standard the since Spring 2003. Mplayer can be set up with menu-support if the *libdvdread*-library is installed.

- OSS applications for mastering of DVD content are still in their childhood. We only found one relevant OSS package for this purpose: *kino*, which still is under development.
- Burning DVD is still complicated, and several incompatibility issues could be discovered. However, when a stable combination of DVD type and DVD drive is found, it works without problems.
- most of the OSS software for multimedia has a massive lack of integration. Often command-line interfaces are found, and integrated applications (like *kino*) are seldom. There is also a greater lack of functionality, and several projects and applications are in beta state.

3.2 Experiences with the distribution

While preparing the **mediAkit** distribution we had success on some fields, and failures on others. Today, **mediAkit** is available on `oss.nr.no`, and will be updated when the need is there.

- Preparation and maintenance of a distribution is an ongoing and time consuming process, which needs a commitment as long as the distribution shall survive. New developments, software updates, and new technical developments must be monitored and appropriate actions must be taken in order to keep the distribution up to date.
- The technical parts of packaging binary distributions is solved with *apt*, *rpm*, and *apt-rpm*, and it works well. The partial distribution of source code would be desirable; however, several technical issues, including missing resolution of source code dependencies, make an upgrade of PMS software necessary. Such an upgrade would be possible to apply, but would be a major effort.²¹
- We decided to use RedHat 8.0 as a basis installation for our work. However, there are many other flavours of Linux available, and the maintenance of packages for all of them would exceed our capacity.
- After announcing the availability of **mediAkit** we hoped on help from the open source community to extend the distribution to several use cases, and other basis distributions. While the software was downloaded, we did not get any response. This observation is similar to the observations of commercial open source software companies, who get response in form of error messages, but rarely source code contributions. More activity on the `www.oss.nr.no` would probably make contribution of software from the outside more interesting.
- Some activities were started to create a Debian-based **mediAkit** distribution (see also Section H). However, we still need the help of the OSS community to perform this task.

Experiences from Alpha-testing. Several candidates were asked to test **mediAkit** in order to find bugs and show where the product could be made better. Unfortunately, we did not have the OSS community to help us in this first phase. See Section I for more details.

3.3 Experiences with NR OSSC

After a first start with *postnuke*, we installed *gforge* in summer 2003. This installation at `www.oss.nr.no` contains a portal with general information, discussion groups, download facilities and cvs repository.

After announcing **mediAkit** in `freshmeat.net` at the beginning of august 2003, we got three days of many hits, and several downloads. After that period an average of about three visitors per day can be observed on the server, and some few downloads of the software. An announcement at `www.nordicos.org` increased the number of visitors slightly. The usage graph for the month of October 2003 is shown in Figure 3.3 as an example for the traffic situation on the server.

Until today we observed only one external user registration in the *gforge* installation; we observed no external contributions to the discussion groups. Obviously the project could not yet gather a community. Unfortunately, a critical mass of stable users and supporters is necessary in order to create a stable and living community around a project.

²¹We took a closer look at the implementation of *apt-rpm*, but the changes would have been too big for the frame of the current project.

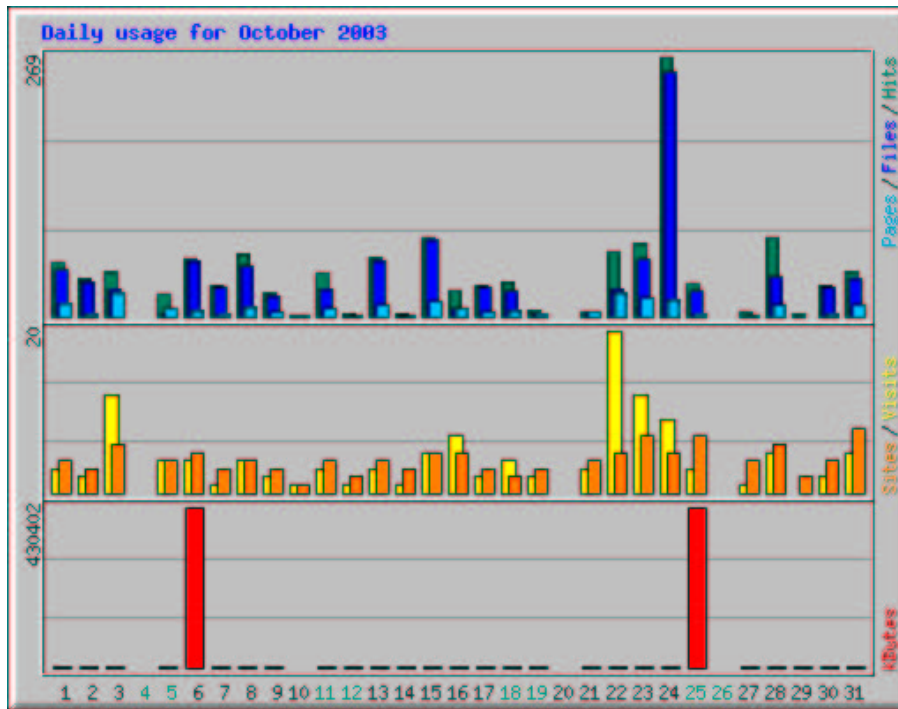


Figure 2: Daily usage graph for October 2003. The graph shows a low, but constant number of visitors. The peaks in the lowest chart shows download of ISO images. The statistics includes hits from our own domain, and hits from search engines on the Internet.

3.4 mediAkit and the Future

The **mediAkit** distribution can only be living as long as there is activity in the project, which is dependent whether funding is available for supervising the project. It is also vital that the project gets contributions from a community. Therefore community building is part of such an enterprise.

Recently, Red Hat announced that the official support for Red Hat 9 and other Red Hat lines will be dropped from April 2004. A new distribution, Fedora will be available as a test bed for Red Hat. Other movements in the Linux market include Novell buying SuSE, and IBMs commitment to Open Source and cooperation to Novell. Therefore, it is necessary to develop **mediAkit** for SuSE Linux, Debian, and other distributions. However, since all of these basis distributions have different profiles, the **mediAkit** project needs support for developing the use cases for these distributions.

Part II

The software catalogue

This part of the report contains an overview over various multimedia program packages for Linux. The list is not exhaustive. We have tried to list the projects that are most useful for end users and most likely to be included in our distribution.

A Tools for building and distribution of software

This section contains a short description of relevant software for the various tasks that have to be solved within the setup project. We intended to keep the descriptions short. However, in order to support the internal discussion regarding some problem domains and its solutions a more profound description was emerging for some areas.

A.1 Build systems

A build system supports how to build executable code out of source code. For our purposes the discussion is limited to tools like make, and how to produce portable makefiles.

A.1.1 make (GNU make)

Package:	make	Version:	3.80	Date:	December 16, 2002
Exports:		Type:		License:	Control file generation
Uses:		License:			GPL
Url:	http://www.gnu.org/software/make/make.html				
Author(s):	Richard Stallman and Roland McGrath (FSF)				

GNU make is a tool which controls the generation of executables and other non-source files of a program from the program's source files. A make utility is normally provided along with C compilers. The concept of *make* goes back to S. Feldman in 1979 [9].

make gets its knowledge of how to build your program from a so-called *makefile*, which is a text file containing a set of named *targets* that can be built. For each target a list of other targets or files that this target depends upon can be listed, followed by a set of command lines that must be executed in order to build this target. Often, a target is the name of a library or a file. Some targets are very often used by convention, e.g., "all" or "install".

When building a target, *make* checks recursively through the list of dependent targets, and performs build actions when necessary, e.g., a newer version of a target is available. In case a library target that a program depends upon has changed, the program has to be rebuilt in order to work with the new library version. If a source file that the library depends upon has changed too, the library has to be rebuilt first.

The *make* system provides functionality for using macros and general rules. If, for instance, the name of the source and output directories are defined using macros, these can easily be redefined without having to rewrite all the target-commands in the *makefile*.

For more information on *make* we refer to the book on GNU Make [24], the documentation in Texinfo [4] format and the manual page.

A.1.2 GNU Autotools

FSF has created a set of tools to support the development and distribution of software across Unix systems. The list of tools includes **autoconf**, **autoheader**, **automake**, **aclocal** and **libtool**.

The use of these tools is a two phase process. In the first phase a software package is prepared for distribution, while in the second phase this package is configured for use on the target system, built and installed. The complexity of the first phase is to ensure maximum flexibility and simplicity for performing the second. The purpose of the first phase is to create (at least) two files: *Makefile.in* and *configure*. The file *configure* is a shell script program that checks for existence, placement

and features of tools, libraries and installed hardware and software that the build process depends on. Thereafter, it configures a set of appropriate `makefiles` from the templates in `Makefile.in`.

autoconf

Package:	autoconf	Version:	2.57	Date:	December 16, 2002
Exports:		Type:		Type:	Configure source packages
Uses:	m4, Perl	License:		License:	LGPL
Url:	http://www.gnu.org/software/autoconf/autoconf.html				
Author(s):	Free Software Foundation (FSF)				

autoconf is an extensible package of m4 macros that produce shell scripts to automatically configure software source code packages. These scripts can adapt the packages to many kinds of UNIX-like systems without manual user intervention. *autoconf* creates a configuration script for a package from a template file that lists the operating system features that the package can use, in the form of m4 macro calls.

Producing configuration scripts using *autoconf* requires GNU *m4* (version 1.4 or later). The configuration scripts produced by *autoconf* are self-contained. Therefore, for the configure-process on the target machine neither *autoconf* nor GNU *m4* need to be installed. Some optional utilities of *autoconf* use *perl*.

autoscan. The **autoscan** program helps to create and/or maintain a `configure.in` file for a software package. *autoscan* examines source files in the directory tree rooted at a directory given as a command line argument, or the current directory if none is given. It searches the source files for common portability problems and creates a file `configure.scan` which is a preliminary `configure.in` for that package, and checks a possibly existing `configure.in` for completeness.

aclocal. **aclocal** is a utility that can provide the file `aclocal.m4` by copying the definition of m4-macros used in `configure.in` from m4-files found on the system. This file is optional input to *autoheader* to help configuring the software for compilation on different platform.

autoheader. **autoheader** is a tool for automatically generating the `config.h.in` file, compliant with the GNU Coding Standards. This file can be used as input to the *configure* script in order to provide cross-platform compatibility.

automake

Package:	automake	Version:	1.7.2	Date:	December 16, 2002
Exports:		Type:		Type:	Generate 'Makefile.in'
Uses:	autoconf	License:		License:	GPL
Url:	http://www.gnu.org/software/automake/automake.html				
Author(s):	Free Software Foundation (FSF)				

automake is a tool for automatically generating `Makefile.in` files compliant with the GNU Coding Standards. This is the basis for the `Makefile` that is output by the *configure* shell script program.

libtool

Package:	libtool	Version:	1.4.3	Date:	December 16, 2002
Exports:		Type:		Type:	Library support script
Uses:		License:		License:	GPL
Url:	http://www.gnu.org/software/libtool/libtool.html				
Author(s):	Gordon Matzigkeit, FSF				

libtool is a generic library support script. Libtool hides the complexity of using shared libraries behind a consistent, and portable interface.

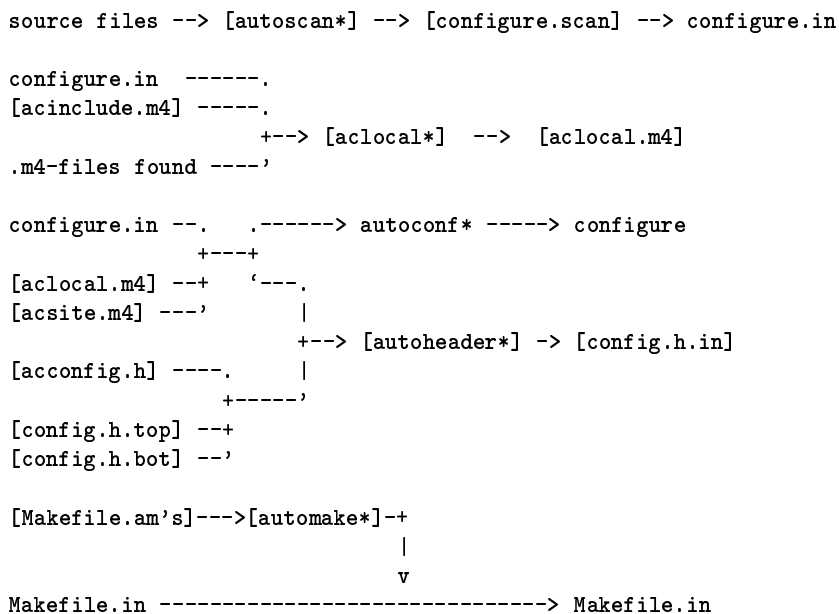


Figure 3: Using autotools.

Using autotools

Preparing a software package with autotools. The use of autotools is quite complex, and involves a number of files and programs. Figure 3 presents a diagram from the autotools manual showing the information flow and program calls when using the autotools. For a more profound description, we refer to the documentation that follows the autotools software package.

Many files may be involved in preparing a software package for distribution. Most of these are optional. Only the files `configure.in` and `Makefile.in` and the use of *autoconf* are mandatory. Using the other tools and files may however ease the maintainability of the software package.

The first step is to run *autoscan* on the source files in order to create the file `configure.in`. The program *aclocal* searches the system for macros in `.m4`-files. Definitions for macros used in `configure.in` are copied to the file `aclocal.m4`.

The two files `configure.in` and `aclocal.m4` are used as input to the command *autoconf*, which creates a shell-script program called *configure*, which is used later to configure the software distribution at the remote site.

The program *autoheader* is used to create the file `config.h`, which contains declarations and compiler switches used in the program. These definitions are included in the program at compile time, which makes it easier to maintain flexibility and compilation across platforms.²²

A template for the `makefile` must be provided in the file `Makefile.in`. In the installation phase the program *configure* will create the appropriate `makefile`. The program *automake* (see Section A.1.2) can be used to maintain a `Makefile.in`.

Configuration and installation of software with autotools. Files and programs used during configuration of a software package are shown in Figure 4. As mentioned above, the shell-script *configure* is generated by the *autoconf* program. The process of configuring, building and installing the program starts by executing this shell-script. *configure* will find the location of directories and system libraries, check for compiler features etc., and save the results in several files. Most of the work is performed by an accompanying script called `config.status`. As a result of this process the appropriate files (e.g., `Makefile`, `config.h`, etc.) are created based on the template files with the extension `.in`.

²²The concept of header files comes from the programming language C, but can be used for other purposes. For that the preprocessor *cpre* can be used.

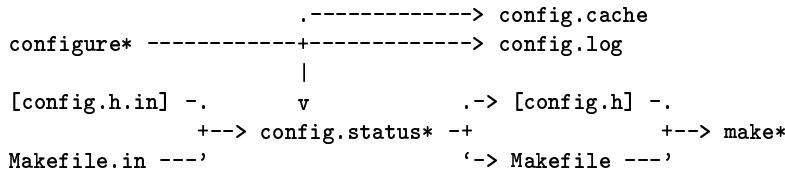


Figure 4: Configuring software.

If the configuring step was successful *make* is run in order to build the software. Finally, *make install* is called to perform the steps necessary to install the newly built parts of the software package.

A.1.3 GAR

Package:	GAR	Version:	CVS	Date:	
Exports:		Type:		Build/install source code	
Uses:	GNU make	License:		BSD	
Url:	http://gar.lnx-bbc.org/				
Author(s):	Nick Moffitt				

GAR is a library of *make* targets that can be configured by setting certain special variables or overridden file-by-file. Each package in the GAR system is contained in a directory inside the directory representing its category. Inside the package directory, there are 2 files: *Makefile* and *checksums*

A.1.4 Ant

Package:	Ant	Version:	1.5.1	Date:	Oct 3, 2002
Exports:		Type:		Build/install source code	
Uses:	Java Runtime or SDK	License:		Apache	
Url:	http://jakarta.apache.org/ant/index.html				
Author(s):	?				

Ant is a platform-independent build tool created by one of the Apache web server developers. The build-specifications are in a platform-independent XML text file. *Ant* is based on the abstraction of “tasks” rather than shell commands, since shell commands are inherently bound to the platform (and sometimes even installation) that they were created on. Dependencies and most other features from *make* are supported. The concept of “task” is more general than the concept of “target” in *make*. Generally, *Ant* is recognised as more versatile and flexible than *make*. *Ant* is a Java program, and needs a Java runtime environment and some Java jar-libraries (for xml processing etc.) in order to run.

A.2 Package Management systems

A.2.1 Dpkg

Package:	Dpkg	Version:	1.9.21 (stable)	Date:	
Exports:		Type:		Package Management System	
Uses:	Perl	License:		GPL	
Url:	http://packages.debian.org/stable/base/dpkg.html				
Author(s):	Debian developers				

Dpkg is the package maintenance system for Debian GNU/Linux. The package format is defined in the Debian Policy manual.

A.2.2 RPM

Package:	RPM	Version:	4.1 (stable)	Date:	
Exports:		Type:		Package Management System	
Uses:	libpopt	License:		GPL	
Url:	http://www.rpm.org/				
Author(s):	Red Hat, Inc.				

RPM is a command-line package management system. It is capable of installing, uninstalling, verifying, querying, and updating software packages. Each software package consists of an archive of files along with information about the package, e.g., version, textual description, etc. The RPM-4.1 format support cryptographic verification of source package builders, using OpenPGP V3 packets. Currently, RPM is being used in RedHat, Mandrake and other distributions. There are also various package managers that support RPM.

Internally, RPM uses the `cpio` format, which is used to archive files in Unix systems (see `man cpio` on a Unix system).

The RPM Package Manager is capable of archiving pre-compiled binaries, as well as compressed source code with binary and text patch files, in a single, common file format archive (`cpio`).

RPM can be used to build, install, query, verify, update and erase individual software packages. RPM works with a database of installed software and files, which maintains dependencies and requirements between installed and upgraded software packages. The RPM database supports queries for files and package details.

RPM supports cryptographic AI validation of the source of packages, by the usage of OpenPGP compatible digital signatures in packages supported by tools, such as the GNU Privacy Guard (GPG).

See Section 16 in the Linux Standard Base Common Specification [26] for the definition of the RPM File format. The command line tool `rpm` is used to manage existing packages, while `rpmbuild` is used to create them. See the manual pages for `rpm(8)` and `rpmbuild(8)` for more details.

Working with RPM

There are three different forms of RPM-packages, binary RPMs, source RPMs (SRPMs) and development RPMs. The naming convention of RPM files follows the following rule:

`<pkg-name>-<version>-<release>.<type>.rpm`

where `<pkg-name>` is the unique name of the software package, `<version>` is a 3-digit major-minor-micro version number, `<release>` is the release number of this version and `<type>` is an identifier for the package type. Valid RPM package types include

- `src`, `nosrc`
- `<architecture>` (`noarch`, `i386`, ..., `i686`, ...)

The entire process of creating and building RPM packages is bound to a strict directory structure:

- `BUILD`
- `RPMS`

- SOURCES
- SPECS
- SRPMS

The locations of these directories can be overridden by passing `--define` options to *rpmbuild* tool. See Section E for an example. The content, building and installation of the package is controlled by a text file called the `spec-file`, usually placed in the `SPECS` directory. The file describes

- Library name, version and vendor
- Dependencies on other packages
- Files (executables and libraries) contributed to the system by installation of the package.
- Dependencies for building (normally development packages with header files etc.)
- Archive(s) that contain the source files, patches etc. needed to build the package.
- The set of commands that must be executed in order to perform preparation, to perform building, installation, cleanup etc.

The program *rpmbuild* can be used to build a Source RPM (SRPM) from source file archives in the `SOURCES` directory and a `spec-file`. It can also build binary RPMs from an SRPMs. The SRPM contains all the necessary source files and the `spec file`. A RPM contains compiled and linked versions of all the libraries and programs in the package, as well as the `spec file`.

It is widespread, but not mandatory, to use *autoconf* / *autotools* and *make* for controlling the build process. RPM is built on top of a database. Queries are sent to the database in order to find out whether dependent packages are installed.

A.2.3 ESP Package Manager (EPM)

Package:	EPM	Version:	3.5.1	Date:	December 18, 2002
Exports:		Type:		Package Management System	
Uses:		License:		GPL	
Url:	http://www.easysw.com/epm/				
Author(s):	Easy Software Products				

EPM is a free UNIX software/file packaging program that generates distribution archives from a list of files. EPM generates portable script-based distribution packages complete with installation and removal scripts, and vendor distributions in AIX, BSD, Compaq Tru64, Debian, HP-UX, IRIX, MacOS X, Red Hat, and Solaris formats.

A.3 Install and Update systems

A.3.1 Loki Setup Graphic Installer

Package:	Loki Setup Graphic Installer	Version:	1.5.8	Date:	
Exports:		Type:		Graphical installer	
Uses:	libxml 1.4.0	License:		GPL	
Url:	http://www.lokigames.com/development/setup.php3				
Author(s):	Sam Lantinga				

The **Loki Setup Graphic Installer** allows the developer to create a simple, easy-to-use, standardised install routine. It requires Loki SetupDB which can be installed from the same site. The installer requires libxml 1.4.0 to parse XML configured description of the files to be installed, and libglade 0.7 to dynamically load the GTK+ user interface definitions.

A.3.2 Loki Update Tool

Package:	Loki Update Tool	Version:	1.0.13	Date:	December 20, 2002
Exports:		Type:		Software Update Tool	
Uses:		License:		GPL	
Url:	http://www.lokigames.com/development/loki_update.php3				
Author(s):	Loki Software, Inc.				

The **Loki Update Tool** is written by Loki Software, Inc., designed to be used in conjunction with their setup and patch tools to easily update products.

This end-user tool allows to quickly and easily update Loki software that is installed already. It features easy-to-use mirror support, the ability to resume broken connections, support for HTTP and FTP proxies, and automatic update verification to make sure that the correct download is performed.

A.3.3 Loki Patch Tool

Package:	Loki Patch Tools	Version:	1.0.1	Date:	December 20, 2002
Exports:		Type:		Binary Patch Tool	
Uses:	Loki SetupDB, xdelta	License:		GPL	
Url:	http://www.lokigames.com/development/loki_patch.php3				
Author(s):	Loki Software, Inc.				

The **Loki Patch Tool** is a set of tools written by Loki Software, Inc., designed to create a set of binary patches that can be easily used in conjunction with the Loki Update Tool.

A.3.4 Red Hat Network Up2date

Package:	Red Hat Network Update Agent	Version:	3.0.7	Date:	
Exports:		Type:		Software maintenance	
Uses:		License:		GPL	
Url:	http://rhn.redhat.com				
Author(s):	Red Hat, Inc.				

The **Red Hat Update Agent** features a GNOME interface for installation and updates of RPM packages from Red Hat and specified software channels.

A.3.5 up2date

Package:	up2date	Version:	2.7.61	Date:	
Exports:		Type:		Software maintenance	
Uses:		License:		GPL	
Url:	http://www.redhat.com/docs/manuals/RHNetwork/ref-guide/up2date.html				
Author(s):	Red Hat, Inc.				

up2date uses the Red Hat Network, and is free for the first computer, but if we want to distribute our own packages we need the server. Red Hat has not released this server, but freely available servers such as Current and nrh-up2date exists.

A.3.6 Current

Package:	Current	Version:	1.4.1	Date:	
Exports:		Type:		Software maintenance	
Uses:		License:		GPL	
Url:	http://www.biology.duke.edu/computer/unix/current/				
Author(s):	Hunter Matthews				

Current is a server for Red Hat's up2date, which is a suite of tools (sometimes called the Red Hat Network) for keeping a Red Hat Linux system "up to date" with new security/bugfix RPM's. It also allows you to add completely new rpm's to clients over the network. The tools do a thorough job of handling bootloader configurations, new kernels, packages dependencies and conflicts.

Unfortunately, Red Hat have not released their server, which does not fully address the needs of some users/departments. Thus, Current was born. Current runs under apache. The locations of rpm and srpm files, including so-called channels are defined in `/etc/current/current.conf`. The up2date client gets the address of the used server in `/etc/sysconfig/rhn/up2date`.

A.3.7 NRH-up2date

Package:	NRH-up2date	Version:	1.1	Date:	December 18, 2002
Exports:		Type:		Software maintenance	
Uses:	Perl, Frontier::RPC2, Digest::MD5	License:		GPL	
Url:	http://www.nrh-up2date.org/				
Author(s):	Alex Kramarov				

NRH-up2date is a collection of utilities and instructions to use with Red Hat's up2date client independent of the Red Hat Network.

A.3.8 APT

Package:	APT	Version:	0.5.4	Date:	December 18, 2002
Exports:		Type:		tool	
Uses:		License:		GPL	
Url:	http://packages.debian.org/stable/base/apt.html				
Author(s):	APT Development team				

APT (Advanced Packaging Tool) is a set of tools to support downloading, installation, and handling of package dependencies. Originally, APT was developed for the Debian distribution, but later ported to other distributions. **APT** is the Debian package management facility. In addition to standard dependency checking and installation functions, the APT system is also able to resolve all dependencies recursively and download necessary packages. The command `prompt> apt-get install some-package` will

- search server locations listed in the file `/etc/apt/sources.list` for the package called *some-package*.
- check the dependencies of this package.
- download the package.
- recursively find, download and install packages that *some-package* depend on that are not yet installed in the proper version.
- install *some-package*.

APT can load packages from various source URLs (including ftp: and http:) as long as they follow a specific directory and content structure. APT searches for packages in locations as specified by the file `/etc/apt/sources.list`.

A.3.9 APT-RPM

Package:	APT-RPM	Version:	0.5.4cnc7	Date:	
Exports:		Type:		Software maintenance	
Uses:	rpm	License:		GPL	
Url:	http://freshmeat.net/articles/view/192/				
Author(s):	Conectiva Inc., Alfredo Kojima				

APT-RPM is a port of Debian's apt tools to a RPM based distribution (Red Hat, Conectiva, SuSE etc.). apt-get is an advanced package management utility front-end which allows the user to easily perform package installation, upgrading and removal. Dependencies are automatically handled, and dependent packages are automatically installed if needed. See [13] for a practical overview.

A.3.10 Ximian Red Carpet

Package:	Ximian Red Carpet Client	Version:	2.1.9.1	Date:	
Exports:		Type:		Software maintenance	
Uses:		License:		GPL	
Url:	http://www.ximian.com/products/redcarpet/				
Author(s):	Ximian, Inc.				

Ximian Red Carpet is a software management solution for Linux desktops from Ximian. It provide a software channel with automatic dependency and conflict resolution that makes it easy to install, update and maintain software over the Internet from Ximian, leading Linux distribution providers and a variety of independent software vendors. Specifically designed for organisations running mixed computing environments, Red Carpet handles a broad range of GNU/Linux distributions, including Red Hat, SuSE, Mandrake, Debian and more.

Ximian released version 1.0 of the Red Carpet Daemon (rcd) and Red Carpet command-line client (rc). See <http://developer.ximian.com/projects/rcd/> for details.

A.3.11 InstallBase

Package:	InstallBase	Version:	1.0a2	Date:	December 20, 2002
Exports:		Type:		Graphical installer	
Uses:		License:		BSD license	
Url:	http://installbase.sourceforge.net				
Author(s):	Damon Courtney				

InstallBase InstallBase MPI is a multi-platform GUI installer designed to be completely cross-platform and function on Windows, Macintosh and most versions of UNIX.

A.3.12 Portage

Package:	Portage	Version:	2.0.46r9	Date:	
Exports:		Type:		Software maintenance	
Uses:	Python 2.2	License:		GPL-2	
Url:	http://www.gentoo.org/dyn/pkgs/sys-apps/portage.xml				
Author(s):	Gentoo Linux				

Portage is a python-based package building system that Gentoo Linux is built around. Gentoo Linux is a source-based Linux distribution. Portage consists of many tools. The most important is maybe emerge, that installs and builds the selected package, and all its dependencies. The Portage repository consists of the source-code in tar.gz-format, and ebuild-scripts that specify how to build the package. According to the Gentoos homepage Portage supports "fine-grained package management, "fake" (OpenBSD-style) installs, path sandboxing, safe unmerging, system profiles, virtual packages, config file management, and more."

A.3.13 Sorcery

Package:	Sorcery	Version:	1.0-beta	Date:	
Exports:		Type:		Software maintenance	
Uses:		License:		SPL/GPL-2	
Url:	http://sorcerer.wox.org/				
Author(s):	Kyle Sallee				

Sorcery is a software maintenance package that was originally developed for the source-based Sorcerer Linux distribution. The project forked later, and the tools therefor exists in 3 different versions, for Sorcerer Linux, SourceMage Linux (<http://www.sourcemage.org/>) and Lunar Linux (<http://www.lunar-linux.org/>). The package contains scripts for installing, uninstalling and bookkeeping of installed software. The package is able of handling dependencies.

A.4 Hardware Detection systems

A.4.1 harddrake

Package:	harddrake	Version:	0.9	Date:	2001
Exports:		Type:		Hardware Detection	
Uses:	GTK+ 1.2, detect	License:		GPL	
Url:	http://www.mandrakelinux.com/harddrake/				
Author(s):	Mandrake, Inc				

HardDrake is a hardware configuration tool written in GTK+ for the Mandrake distribution.

A.4.2 detect

Package:	detect	Version:	0.9.72	Date:	2001
Exports:		Type:		Hardware Detection	
Uses:		License:		GPL	
Url:	http://www.mandrakelinux.com/harddrake/				
Author(s):	Mandrake, Inc				

Detect is a library for automatic hardware detection. The following hardware can be detected: CPU, memory, disks, web cams, joysticks, partitions, Ethernet cards, floppy drives, modems, mice, SCSI, sound cards, and video cards.

A.4.3 kudzu

Package:	kudzu	Version:	0.99.69 (CVS)	Date:	October 18, 2002
Exports:		Type:		Hardware Detection	
Uses:	pciutils python newt	License:		GPL	
Url:	http://rhlinux.redhat.com/kudzu/				
Author(s):	Red Hat, Inc.				

kudzu is the Red Hat Linux hardware probing library, and the associated configuration program. The probing library is used by various system utilities, such as anaconda, Xconfigurator, and hwbrowser. The tool runs at system boot time to determine what hardware has been added or removed from the system.

A.4.4 discover

Package:	discover	Version:	2.0.1	Date:	October 24, 2002
Exports:		Type:		Hardware Detection	
Uses:	libdiscover	License:		GPL	
Url:	http://archive.progeny.com/progeny/discover/				
Author(s):	Progeny, Inc.				

Discover is a cross-platform hardware detection system that uses system-dependent modules (selected at build time) for detecting the hardware on a system. It also provides system-independent interfaces for querying XML data sources about this hardware.

A.4.5 Kernel Autoconfigure

Package:	Kernel Autoconfigure	Version:	1.9	Date:	December 16, 2002
Exports:		Type:		Hardware Detection	
Uses:	cml2	License:		GPL	
Url:	http://sourceforge.net/projects/kautoconfigure/				
Author(s):	Giacomo Catenazzi				

Kernel Autoconfigure is a tool for configuring the Linux kernel, detecting hardware and used software protocols.

A.4.6 libhardware

Package:	libhardware	Version:	0.7.1	Date:	November 19, 2002
Exports:		Type:		Hardware Detection	
Uses:		License:		GPL	
Url:	http://www.badpenguin.org/				
Author(s):	Antonio Gallo				

libhardware is a static library used in many utilities that are part of the AGX's GNU/Linux distribution "Bad Penguin". It provides functions to access information about the hardware of the system.

A.5 Bug Tracking systems

A.5.1 Bugzilla

Package:	Bugzilla	Version:	2.17.1	Date:	December 20, 2002
Exports:		Type:		Type:	Bug Tracking system
Uses:	Apache	License:		License:	MPL
Url:	http://www.mozilla.org/projects/bugzilla/				
Author(s):	Dave Miller				

Bugzilla is a web-oriented database for reporting problems (bugs) in software. It lets people report bugs and assigns these bugs to the appropriate developers. Developers can use Bugzilla to keep a to-do list as well as to prioritise, schedule and track dependencies.

A.5.2 PHP bugtracker

Package:	PHP bugtracker	Version:	0.9.0	Date:	December 20, 2002
Exports:		Type:		Type:	Bug Tracking system
Uses:	Apache, PHP	License:		License:	GPL
Url:	http://phpbt.sourceforge.net/				
Author(s):	Benjamin Curts				

PHP bugtracker is designed as a replacement for Bugzilla. This project grew out of the frustrations experienced in installing and using Bugzilla. Design goals for PHP bugtracker are simplicity in use and installation, use of templates to achieve presentation independence, and the use of a database abstraction layer for independence from a data base.

A.5.3 Bug-buddy

Package:	Bug-buddy	Version:	2.2.0	Date:	December 20, 2002
Exports:		Type:		Type:	Bug Tracking System
Uses:	GNOME, Bugzilla	License:		License:	GPL
Url:	ftp://ftp.gnome.org/pub/GNOME/stable/sources/bug-buddy/				
Author(s):	Jacob Berkman				

Bug-buddy is an interface tool for Bugzilla. Features include the ability to obtain a stack trace from a core file or crashed application and to determine versions of packages installed on a system.

A.5.4 GNATS

Package:	GNATS	Version:	3.999.2 (stable)	Date:	December 20, 2002
Exports:		Type:		Type:	Bug Tracking System
Uses:		License:		License:	GPL
Url:	http://sources.redhat.com/gnats/				
Author(s):	GNATS development Team				

GNATS contains a database and tools for administering problem reports. The data base is file based. Most of the administration tools are based on formatted messages. Gnats includes administration tools, a server program and various tools to access the information by the end user, e.g., interfaces towards email, emacs, and web.

A.6 Software Image systems

A software image system is capable of building (bootable) install images from a tree of software packages. For our purposes the discussion is limited to building and recording a image of the software distribution packages onto CDROM and DVD media.

A.6.1 BICK

Package:	BICK	Version:	0.8.0	Date:	December 16, 2002
Exports:		Type:		Type:	ISO image generation
Uses:		License:		License:	GPL
Url:	http://phil.ipal.org/freeware/bick/				
Author(s):	Philip Howard				

BICK builds a bootable Linux ISO image from a file tree that represents the run time file tree loaded into tmpfs with options to leave /opt and /usr mounted on the CDROM to save space. For systems with enough RAM, up to as much file data as will fit on the CDROM can be loaded; initial ramdisk is not used for the runtime files.

A.6.2 Progeny Graphical Installer

Package:	PGI	Version:	1.0.1	Date:	December 16, 2002
Exports:		Type:		Type:	ISO image generation
Uses:		License:		License:	GPL
Url:	http://archive.progeny.com/progeny/pgi/				
Author(s):	Progeny				

Progeny Graphical Installer (PGI) is a multi-architecture system for creating graphical installers for installing a bootable Debian GNU/Linux operating system onto a new computer system.

The PGI package enables the user to create ISO images containing a bootable installer that guides the user through the steps of installation. The installer supports text and graphical installation modes. ISO images can be generated with complete or partial package archives, or with the installer only (useful for network-only installs, which PGI supports).

A.6.3 Anaconda

Package:	Anaconda	Version:	CVS	Date:	December 16, 2002
Exports:		Type:		Type:	ISO image generation
Uses:	python	License:		License:	GPL
Url:	http://rhlinux.redhat.com/anaconda/				
Author(s):	Red Hat, Inc.				

Anaconda is the Red Hat Linux Installation Program. It supports installation media tests, and several installation methods.

A.7 Web portal and content management systems

A.7.1 Apache

Package:	Apache	Version:	1.3.27	Date:	December 18, 2002
Exports:		Type:		Type:	HTTP/1.1 Server
Uses:		License:		License:	Apache Software License
Url:	http://httpd.apache.org/				
Author(s):	Apache Software Foundation				

The **Apache** HTTP Server Project is an effort to develop and maintain an open-source HTTP server.

A.7.2 Plone

Package:	Plone	Version:	1.0 RC1	Date:	December 20, 2002
Exports:		Type:		Type:	Content Management
Uses:	Apache, Zope		License:	BSD	
Url:	http://www.plone.org/				
Author(s):	Alexander Limi, Alan Runyan, Vidar Andersen				

Plone is built on top of the open source application server Zope (see <http://www.zope.org/>) and the accompanying Content Management Framework. Plone can be used as an intranet server, as a document publishing system and as a groupware tool for collaboration between separately located entities. Plone currently supports approximately 20 languages.

A.7.3 Postnuke

Package:	Postnuke	Version:	0.7.2.3	Date:	December 20, 2002
Exports:		Type:		Type:	Content Management
Uses:	Apache, PHP		License:	GPL	
Url:	http://www.postnuke.com/				
Author(s):	PostNuke development team				

Postnuke was originally a fork of PHPNuke by Francisco Burzi. Many of the concepts of PHPNuke survived even though the code is rewritten. Postnuke is also influenced by earlier products such as *slash* and *thatware*.

Many open source Content Management solutions exist, and choosing the right one is often a subjective decision.

A.7.4 GForge

Package:	GForge	Version:	3.0	Date:	May 11, 2003
Exports:		Type:		Type:	Content Management
Uses:	Apache, PHP		License:	GPL	
Url:	http://gforge.org/				
Author(s):	Tim Perdue, et.al.				

GForge is an Open Source collaborative software development tool, which allows to organise and manage software development projects. It supports managing large teams of software engineers and/or engineers scattered among multiple locations. GForge uses a Web interface, and includes tools like Bug tracker, patches, feature requests, support requests, discussion forum, and project management tools.

GForge is a fork of the 2.6.1 SourceForge code, which was available via anonymous CVS from VA (Research Linux Software). The fork was made after VA did not release their code for a while. GForge 3.0 is a further development including new features.

B Open Source Multimedia Software

Quite a lot of interesting software exist for Linux systems. Until recently, Linux distributions had only modest amount of multimedia software included. This may be influenced by the facts that much of this software is officially pre-release, the software development projects are moving quickly, that the set of dependencies between the various projects is quite complex and that some of the software is hardware-dependent.

This part of the report contains an overview of relevant multimedia software. We are aware that there are many other software packages out there, that would fit into the categories mentioned in our software catalogue. Open source software on digital TV is part of another project within the Channel S program, and therefore not listed here. Readers interested in open source projects on digital TV are referred to the project report of the digital TV project [15].

B.1 Frameworks and multitools

B.1.1 GStreamer

Package:	GStreamer	Version:	0.5.0	Date:	December 16, 2002
Exports:	most available formats			Type:	framework
Uses:	Gnome widgets, external tools, e.g., Icecast			License:	GPL
Url:	http://gstreamer.net/				
Author(s):	Erik Walthinsen, (many others)				

GStreamer is a framework similar to Microsoft DirectShow (aka. ActiveMovie). Can handle (input and output) many file types. Many plugins exist, including for Icecast. However, there is no support for Win32 plugins. A command line utility, graphical graph editor and multimedia SDK are included in the same package.

B.2 Capturing

B.2.1 dvgrab

Package:	dvgrab	Version:	1.2	Date:	February 23, 2003
Exports:				Type:	Video capture
Uses:	libraw1394, 1394-compatible kernel			License:	GPL
Url:	http://kino.schirmacher.de/				
Author(s):	Dan Dennedy				

dvgrab is an utility to capture video from a DV camera. It will capture digital video from a DV source on the IEEE-1394 bus.

B.2.2 xawtv

Package:	xawtv	Version:	3.85	Date:	February 3, 2003
Exports:				Type:	TV application
Uses:				License:	GPL
Url:	http://bytesex.org/xawtv/				
Author(s):	Gerd Knorr				

xawtv is a suite containing different applications for the video4linux interface. It contains among other things a tv-watching application, and *streamer*, a command line tool for capturing video and still images.

B.2.3 Coriander

Package:	Coriander	Version:	0.27.1	Date:	November 26, 2002
Exports:		Type:		Type:	Digital camera control
Uses:	GTK+ >= 1.2, libraw1394, libdc1394, 1394-compatible kernel		License:	GPL	
Url:	http://www.tele.ucl.ac.be/PEOPLE/DOUXCHAMPS/ieee1394/coriander/				
Author(s):	Damien Douxchamps				

Coriander is a GUI for Linux to control a digital camera through the IEEE1394 bus (aka. FireWire, or iLink). The digital camera must comply with the IIDC v1.04 (or later) Digital Camera Specifications, published by the 1394 Trade Association.²³ We did not test coriander, due to compilation problems with *libdc1394*.

B.2.4 libgphoto2

Package:	libgphoto2	Version:	2.1.1	Date:	November 2002
Exports:	JFIF, TIFF, EXIF	Type:		Type:	Digital camera library
Uses:	libusb	License:	LGPL		
Url:	http://gphoto.sourceforge.net/				
Author(s):	Scott Fritzing, Lutz Müller, et.al.				

The **libgphoto2** library is used by applications to access and control various digital still camera models. A command-line frontend follows with the library. Several graphical front ends for the library are available.

B.2.5 gtkam

Package:	gtkam	Version:	0.1.10	Date:	November 2002
Exports:	JFIF, TIFF, EXIF	Type:		Type:	Digital photo application
Uses:	libgphoto2 >= 2.1.1, GTK+ >= 2.0		License:	GPL	
Url:	http://gphoto.sourceforge.net/				
Author(s):	Scott Fritzing, Lutz Müller				

gtkam provides a GTK+ front-end to libgphoto2.

B.2.6 digiKam

Package:	digiKam	Version:	0.5.1	Date:	December 20, 2002
Exports:		Type:		Type:	Digital photo application
Uses:	libgphoto2, KDE		License:	GPL	
Url:	http://digikam.sourceforge.net/				
Author(s):	Renchi Raju				

digiKam is a KDE application which acts as a frontend to gphoto2 for communicating with a digital camera and download pictures from it.

B.2.7 SLab

Package:	SLab	Version:	4.09b	Date:	December 20, 2002
Exports:		Type:		Type:	Multitrack HD recording
Uses:		License:	Partly GPL		
Url:	http://slabexchange.org/				
Author(s):	Nick Copeland, Toby Shepard				

²³Some video tape operated cameras (i.e. camcorders, etc.) are not compliant with the IIDC specifications. Those cameras record compressed DV video on a tape, while the IIDC protocol is specified for uncompressed, on-the-fly video flow.

SLab is a digital audio recording software suite for UNIX platforms, which is currently supported for Linux, and FreeBSD.

SLab can record from up to eight stereo sound cards to files. The audio data can then simultaneously be read back from disk as separate tracks, and mixed down back to the audio devices. SLab supports up to 64 tracks.²⁴ Note, SLab is not a real time mixer, so its use is not intended for live work.

Not all parts of SLab are in open source. Most of the APIs are available as source, to access the database, build effects, extend some of the mixing algorithms; additionally all of the audio driver interface code is in the GPL.

B.2.8 Ardour

Package:	Ardour	Version:	0.522.4	Date:	December 2002
Exports:	WAV	Type:		Type:	Multichannel HD recorder
Uses:	ALSA, JACK	License:		License:	GPL
Url:	http://ardour.sourceforge.net/				
Author(s):	Paul Davis				

Ardour is a multichannel hard disk recorder (HDR) and digital audio workstation (DAW). It is capable of simultaneous recording 24 or more channels of 32 bit audio at 48kHz. Ardour is intended to function as a “professional” HDR system, replacing dedicated hardware solutions such as the Mackie HDR, the Tascam 2424 and more traditional tape systems like the Alesis ADAT series. It is also intended to provide the same or better functionality as proprietary software DAWs such as ProTools, Samplitude, Logic Audio, Nuendo and Cubase VST. It supports MIDI Machine Control, and so can be controlled from any MMC controller, such as the Mackie Digital 8 Bus mixer and many other modern digital mixers.

B.2.9 XMMS Recorder

Package:	XMMS Recorder	Version:	0.3.0	Date:	December 20, 2002
Exports:	WAV	Type:		Type:	plugin
Uses:	XMMS	License:		License:	GPL
Url:	http://xmms-recorder.sourceforge.net/				
Author(s):	Tamás Skopkó, Mike Tyson				

XMMS recoder is an XMMS input plugin that records from your soundcard’s input and sends audio to the selected output. It can be used for recording audio to disk, or running a visualisation on an input stream. XMMS Recorder works with OSS, eSounD and ALSA.

B.2.10 LiveIce, LiveIce XMMS

Package:	LiveIce / LiveIce XMMS	Version:	1.0	Date:	2001
Exports:		Type:		Type:	library
Uses:	GTK, GDK, libraw1394, libdc1394, 1394-compatible kernel	License:		License:	GPL
Url:	http://star.arm.ac.uk/~spm/software/liveice.html				
Author(s):	Damien Douxchamps				

LiveIce was developed after the appearance of IceCast to provide the Real Time streaming functionality which makes it a viable solution for broadcasting. **LiveIce XMMS** is an effects plugin for xmms which sends streams to IceCast and provides a limited set of functions similar to those provided by liveice but without half the hassle. LiveIce XMMS lacks many of the features of the full version of LiveIce but provides the main functions needed to stream from xmms.

²⁴However, the CPU power might restrict that; a P133 will max out at about 8 tracks, a P450 has been tested to 16 tracks (even with the more intensive mix algorithms) with capacity to spare.

B.2.11 Jack, Jackit, jackd, libjack

Package:	Jack audio connection kit	Version:	0.34	Date:	2001
Exports:	sound, via low latency callbacks	Type:		License:	library, daemon
Uses:	ALSA, RealTime kernel patch recom- mended				
Url:	http://jackit.sourceforge.net				
Author(s):	Paul Davis, Richard Günter et al.				

JACK is a low-latency audio server, written primarily for the GNU/Linux operating system. It can connect a number of different applications to an audio device, as well as allowing them to share audio between themselves. Its clients can run in their own processes (i.e. as normal applications), or can they can run within the JACK server (i.e., as a “plugin”). The system focuses on two key areas: synchronous execution of all clients, and low latency operation.

B.2.12 gPhoto

Package:	gPhoto	Version:	2.1.0	Date:	November 2002
Exports:		Type:		License:	JFIF, EXIF, TIFF
Uses:					GPL, LGPL
Url:	http://gphoto.sourceforge.net				
Author(s):	Christophe Barbe, Hubert Figuiere, Scott Fritzing, Lutz Müller, Hans Ulrich Niedermann				

gPhoto is a program and library framework that lets users download pictures from their digital cameras. There are currently 253+ supported digital cameras across several platforms.

B.2.13 SANE

Package:	SANE	Version:	1.0.9	Date:	December 20, 2002
Exports:	TIFF, JFIF	Type:		License:	Scanner Library
Uses:					GPL
Url:	http://panda.mostang.com/sane/				
Author(s):	David Mosberger, others				

SANE stands for “Scanner Access Now Easy”. It is an application programming interface (API) that provides standardised access to any raster image scanner hardware (flatbed scanner, hand-held scanner, video- and still-cameras, frame-grabbers, etc.). The SANE API is public domain and its discussion and development is open to everybody. The current source code is written for UNIX (including GNU/Linux). Several front ends to SANE are available.

B.2.14 vsound

Package:	vsound	Version:	0.5.5	Date:	December 20, 2002
Exports:	WAV	Type:		License:	Recording program
Uses:					GPL
Url:	http://www.xenoclast.org/vsound/				
Author(s):	Erik de Castro Lopo				

VSound can be compared with a virtual audio loopback cable. It allows to record the output audio stream of a program. This is similar to the process of connecting a loopback cable to the line-in and line-out jacks on the sound card, and recording the sound from the line-in jack without the DA/AD conversion losses. VSound can be used as part of a RealAudio-to-WAV file converter.

B.3 Video production and editing

B.3.1 Heroine Virtual: Cinelerra

Package:	Heroine Virtual: Cinelerra	Version:	1.1	Date:	December 18, 2002
Exports:		Type:		License:	library
Uses:	xmovie, libmpeg3 mpeg-2-movie, quick-time4linux (all included)				
Url:	http://heroinewarrior.com/cinelerra.php3				
Author(s):	Heroine Virtual Ltd.				

Website: “Heroine Virtual Ltd. presents an advanced composing and editing system for native Linux at no cost to users. Cinelerra is not for consumer use. [...] If ease of use, simplicity, and convenience are your thing, you should use Virtualdub, Kino, MJPEG tools or MainActor instead.” Cinelerra provides Renderfarm support.

B.3.2 Kino

Package:	Kino	Version:	0.6.4	Date:	February 21, 2003
Exports:	Uncompressed video		Type:	Non-Linear Editor	
Uses:			License:	GPL	
Url:	http://kino.schirmacher.de/				
Author(s):	Arne Schirmacher				

Kino is a non-linear DV editor for GNU/Linux. It features integration with IEEE-1394 for capture, VTR control, and recording back to the camera. It captures video to disk in RawDV and AVI format, in both type-1 DV and type 2-DV (separate audio stream) encoding.

B.3.3 EffecTV

Package:	EffecTV	Version:	0.3.8	Date:	December 18, 2002
Exports:	Uncompressed video		Type:	application	
Uses:	video4linux/bttv-driver, SDL, GCC w/NASM, vloopback		License:	GPL	
Url:	http://effectv.sourceforge.net				
Author(s):	Kentarou Fukuchi, Sam Mertens, Matthias Kleinmann, Buddy Smith, Christian W. Zuckschwerdt, Christian Berger, Ed Tannenbaum, Ico Doornekamp, Jun Ilo				

EffecTV offers a compilation of visual effects and a program that applies these in real-time to captured video. Built on existing libraries for capturing content (bttv-driver) and accessing graphics frame buffers (SDL - Simple DirectMedia Layer). Available for PlayStation 2 running Linux. The optional vloopback package lets other capture programs use the processed frames as if they were from the original data source.

B.3.4 Film Gimp

Package:	Film Gimp	Version:	0.11	Date:	December 15, 2002
Exports:			Type:	Motion Picture Editor	
Uses:	GTK+ >= 1.2.10, libtiff, libpng, libjpeg		License:	GPL	
Url:	http://filmgimp.sourceforge.net/				
Author(s):	Robin Rowe, Sourceforge				

Film Gimp is a motion picture editing tool primarily used for painting and retouching of movies. Film Gimp is the most successful open source tool in feature motion picture work today. Film Gimp runs on Linux and SGI Irix. A Windows version is planned for release in December 2002, and Macintosh native in 2003.

B.3.5 Linux Video Studio

Package:	Linux Video Studio	Version:	0.1.7	Date:	October 21, 2002
Exports:		Type:		Type:	Video Editing
Uses:	mjpegtools	License:		License:	GPL
Url:	http://ronald.bitfreak.net/index.shtml				
Author(s):	Ronald Bultje and Bernhard Praschinger				

Linux Video Studio works as a graphical user interface for the different tools in the mjpeg-tools package. It's capable of capturing video from Video4Linux-devices or from Zoran based cards. IEEE1394 is however not supported at the moment, since a Video4Linux driver for IEEE1394 isn't implemented yet. The program is also capable of editing and playback videos. It can also encode the video in different MPEG-formats, or in VCD- or DivX-format.

B.4 Audio production and editing

B.4.1 Audacity

Package:	Audacity	Version:	1.1.1	Date:	December 3, 2002
Exports:		Type:		Type:	Audio editor
Uses:	wxWindows	License:		License:	GPL
Url:	http://audacity.sourceforge.net/				
Author(s):	Dominic Mazzoni				

Audacity is an audio editor which can record sounds, play sounds, import and export WAV, AIFF, and MP3 files, and more. It can be used to edit sounds using Cut, Copy and Paste (with unlimited Undo), mix tracks together, or apply effects to recordings. It also has a built-in amplitude envelope editor, a customisable spectrogram mode and a frequency analysis window for audio analysis applications. Built-in effects include Bass Boost, Wahwah, and Noise Removal, and it also supports VST plug-in effects. Audacity uses wxWindows, a cross platform GUI toolkit, available from <http://www.wxwindows.org>.

B.4.2 Common Lisp Music

Package:	CLM	Version:	2	Date:	December 2001
Exports:		Type:		Type:	Music synthesis
Uses:	Common Lisp environment	License:		License:	Free
Url:	http://www-ccrma.stanford.edu/software/clm/				
Author(s):	Bill Schottstaedt				

Common Lisp Music (CLM) is a music synthesis and signal processing package in the Music V family.

B.4.3 Common Music Notation

Package:	CMN	Version:		Date:	December 2002
Exports:		Type:		Type:	Music notation
Uses:	Common Lisp environment	License:		License:	Free
Url:	http://www-ccrma.stanford.edu/software/cmn/				
Author(s):	Bill Schottstaedt				

Common Music Notation (CMN) is a Lisp package that can create and display traditional western music scores.

B.4.4 DAP

Package:	DAP	Version:	2.1.4	Date:	
Exports:		Type:		Audio processor	
Uses:	XForms	License:		GPL	
Url:	http://www.cee.hw.ac.uk/~richardk/				
Author(s):	Richard Bruce Kent				

DAP is a digital audio processor for Unix, originally developed for SGI. It now supports Solaris, Linux and ships with several distributions of the Linux operating system.

B.4.5 GNUsound

Package:	GNUsound	Version:	0.3.7	Date:	
Exports:		Type:		Audio editor	
Uses:		License:		GPL	
Url:	http://awacs.dhs.org/software/gnusound/				
Author(s):	Pascal Haakmat				

GNUsound is a sound editor for Linux/x86. It supports multiple tracks, multiple outputs, and 8, 16, or 24/32 bit samples. It can read a number of audio formats through libaudiofile, and saves them as WAV.

B.4.6 MUSE

Package:	MUSE	Version:	0.5.2	Date:	
Exports:		Type:		Audio editor	
Uses:		License:		GPL	
Url:	http://muse.seh.de/				
Author(s):					

MusE is a MIDI/Audio sequencer with recording and editing capabilities. Not tested due to problems with installation.

B.4.7 Rezound

Package:	Rezound	Version:	0.4.0beta	Date:	
Exports:		Type:		Audio editor	
Uses:		License:		GPL	
Url:	http://rezound.sourceforge.net/				
Author(s):	Davy Durham				

Rezound ReZound aims to be a stable, open source, and graphical audio file editor primarily for but not limited to the Linux operating system.

B.4.8 TerminatorX

Package:	TerminatorX	Version:	3.73	Date:	December 16, 2002
Exports:		Type:		Audio synthesizer	
Uses:	GTK+ >= 2.0, libaudiofile, libvorbis, mad	License:		GPL	
Url:	http://terminatorx.cx/				
Author(s):	Alexander Koenig				

TerminatorX is a realtime audio synthesizer that allows the user to “scratch” on digitally sampled audio data (*.wav, *.au, *.ogg, *.mp3, etc.) the way hiphop-DJs scratch on vinyl records. It features multiple turntables, realtime effects (buit-in as well as LADSPA plugin effects), a sequencer and an easy-to-use GUI based on GTK+.

B.5 Encoders and Transcoders

B.5.1 sampeg-2

Package:	sampeg-2	Version:	0.6.4	Date:	
Exports:		Type:	encoder		
Uses:		License:	GPL		
Url:	http://rachmaninoff.ti.uni-mannheim.de/sampeg/				
Author(s):	Dirk Farin				

SAMPEG-2 is a software MPEG-1 and MPEG-2 encoder, designed to generate good image quality and provide a flexible framework for the design of new coding algorithms.²⁵

The features include:

- supports parallel encoding on SMP systems with very high speed-ups,
- scene-change detection integrated to provide good reference frames for motion-estimation at scene-changes, and to exploit the temporal masking effect of the human visual system,
- adaptive quantization used to exploit the activity masking effect,
- several standard motion-estimation algorithms are implemented,
- MJPEG-AVI input with audio extraction,
- direct BTTV-device input for real-time encoding,
- MMX (x86) and VIS (UltraSparc) optimised code available.

B.5.2 ffmpeg

Package:	Ffmpeg	Version:	CVS-2002-12-9	Date:	December 10, 2002
Exports:		Type:	Encoder		
Uses:		License:	(L)GPL		
Url:	http://ffmpeg.sourceforge.net/				
Author(s):	Gerard Lantau, Sourceforge				

The libavcodec library, which is a significant part of ffmpeg, provides a lot of codecs, such as RealVideo 1.0, MJPEG, H263, H263+, as well as MPEG-1 (VCD) and MPEG-2 (DVD) video and audio codecs. (mpeg1-video, mpeg-1/2 audiolayer 2, DivX4/5 (mpeg4), MS-mpeg4-v3 (DivX3), H.263+, mjpeg, AC3).

B.5.3 libmpeg2

Package:	libmpeg2	Version:	0.3.0	Date:	November 27, 2002
Exports:		Type:	library		
Uses:		License:	GPL		
Url:	http://libmpeg2.sourceforge.net/				
Author(s):	Sourceforge				

libmpeg2 is a free library for decoding MPEG-2 and MPEG-1 video streams.

B.5.4 MEncoder

Package:	MEncoder	Version:	0.90rc1	Date:	December 7, 2002
Exports:		Type:			
Uses:		License:	GPL		
Url:	http://www.mplayerhq.hu/				
Author(s):	A'rpi/ESP-team, MPlayer community				

MEncoder can encode/transcode from MPlayer-playable movies (codecs include AVI, VCD, VOB, MPG, MOV, VIV, FLI, RM, NUV, NET) to other MPlayer-playable codecs like DivX4 (1 or 2 passes), libavcodec, PCM/MP3/VBRMP3 audio. MEncoder also supports stream copying and video resizing.

²⁵The development of SAMPEG-2 has been set on hold in favour of the SAMPEG-4 project which will also provide an integrated MPEG-4 encoder.

B.5.5 lame

Package:	lame	Version:	3.93	Date:	December 20, 2002
Exports:	lame	Type:	tools, mp3-encoder		
Uses:	lavtools	License:	GPL		
Url:	http://www.mp3dev.org				
Author(s):	Mike Cheng				

LAME originally stood for LAME Ain't an Mp3 Encoder. The reason for this is that it was only a patch for the ISO demonstration mp3-encoder source. In May 2000 the LAME project became a stand-alone mp3-encoder with no dependency of the ISO source. Lame can encode MPEG1/2 Layer 3 audio with constant and variable bitrate.

B.5.6 liba52

Package:	liba52	Version:	0.7.4	Date:	December 10, 2002
Exports:		Type:	library		
Uses:		License:	GPL		
Url:	http://liba52.sourceforge.net/				
Author(s):	liba52 Team, Sourceforge				

liba52 is a library for decoding ATSC A/52 streams. The A/52 standard (also known as AC-3) is used in a variety of applications, including digital television and DVD. The liba52 distribution contains a test program, a52dec. It decodes ATSC A/52 streams, and also includes a demultiplexer for MPEG-1 and MPEG-2 program streams.

B.5.7 libdv

Package:	libdv	Version:	0.98	Date:	July 29, 2002
Exports:		Type:	library		
Uses:		License:	GPL		
Url:	http://libdv.sourceforge.net				
Author(s):	Charles Krasic and Erik Walthinsen, Oregon Graduate Institute of Science and Technology				

Libdv was developed according to the official standards for DV video: IEC 61834 and SMPTE 314M. The Quasar DV codec (libdv) is a software codec for DV video, the encoding format used by most digital camcorders, typically those that support the IEEE 1394 (a.k.a. FireWire or i.Link) interface.

B.5.8 mpegtools

Package:	dvb-mpegtools	Version:	0.9.4	Date:	2002-03-21
Exports:		Type:	applications		
Uses:	code from MPEG	License:	GPL		
Url:	http://people.debian.org/~blade/dvb.html				
Author(s):	Eduard Bloch, Markus Metzler (2001), Christoph Moar (1995)				

mpegtools are a variety of tools that transform between various MPEG stream formats. The package contains applications for conversion between several MPEG stream types: TS, PES, and ES. A special format for the Siemens DVB card is also included. Additionally there are tools to retrieve characteristics and type of a stream.²⁶

The functions include:

- streamtype (information about stream)
- ts2pes (transforms a transport stream into an MPEG2 program stream)
- ts2av_pes transforms a TS into the AV_PES format used by the Siemens dvb card.

²⁶mpegtools is included in the HIOF DVB package. It is also part of the dvb Debian package.

- `pes2aud`, `pes2vid`, `pes2aud_es`, `pes2vid_es`: extract audio or video streams from PES or PS streams depending on their respective names. The `_es` part of the name indicates that elementary streams are extracted in contrast to PES streams.
- `av_pes2aud`, `av_pes2vid_es`: same as above but for AV_PES streams.
- `pes2av_pes`, `ps2av_pes`, `av_pes2ps`, `av_pes2pes`, `av_pes2ts`: transform AV_PES in PS, PES and TS, respectively.
- `check_av_pes`: Gives some information about the AV_PES packets while outputting the AV_PES stream.
- `tspids`: return PIDs available in a stream.
- `es2av_pes`: turns a video ES into AV_PES.
- `pes2ts2`: transforms a PES stream into a TS (muxed).
- `analyze`: gives information about a PS/PES stream and all the packets contained therein.
- `pes_demux`, `es_demux`: demuxes a PS/PES into separate audio and video streams.
- `ts_demux`: demuxes a TS into PES.
- `ts_es_demux`: demuxes a TS into PES.
- `ts2es`: demuxes a TS into ES, but only for the given PID.
- `pesplot`: returns the audio and video PTS of the PES packets in a gnuplot plattable format.
- `remux`: is the first try of a program to remux a PS/PES stream.
- `pes_repack`: repacks your PES packets into smaller sizes.
- `mplex`: the main part of `mplex` is a rewrite of the multiplexing routines of `bbmpeg`, (`mplex` by Christoph Moar).

B.5.9 transcode

Package:	<code>transcode</code>	Version:	0.6.2	Date:	December 10, 2002
Exports:		Type:		Type:	tools
Uses:		License:		License:	GPL
Url:	http://www.theorie.physik.uni-goettingen.de/~ostreich/transcode/				
Author(s):	Thomas Östreich				

`transcode` is a text console video-stream processing tool that supports elementary video and audio frame transformations. Some example modules, e.g. `ffmpeg` (see B.5.2), are included to enable import of MPEG-1/2, Digital Video, and other formats. It also includes export modules for writing to AVI files with DivX, OpenDivX, XviD, Digital Video or other codecs. Direct DVD transcoding is also supported. A set of tools is available to extract and decode the sources into raw video/audio streams for import and to enable post-processing of AVI files.

B.5.10 mjpegtools

Package:	<code>mjpegtools</code>	Version:	1.6.1	Date:	December 10, 2002
Exports:	<code>lavtools</code> , <code>xlav</code> , <code>mplex</code> , <code>aenc</code> , <code>mjpeg</code>	Type:		Type:	tools
Uses:	<code>lavtools</code>	License:		License:	GPL
Url:	http://mjpeg.sourceforge.net/				
Author(s):	Gernot Ziegler, Ronald Bultje, Bernhard Praschinger, Andrew Stevens				

The `mjpegtools` package provides Linux Audio and Video tools Motion-JPEG and MPEG. It implements MJPEG recording and playback, and simple cut-and-paste editing as well as MPEG compression of audio and video. The code has partially been written by the `mjpegtools` authors, but also code from various other open source projects is included. The package consists of the following parts / directories:

- **lavtools, xlav and utils:** contains a version of `lavtools` by Rainer Johanni, to handle AVI and Quicktime MJPEG files. The package contains applications to record, encode, replay and decode these files, user interface components, and some conversion tools for adding and extracting components (`lav2yuv`, `lav2wav`). The package also includes image processing

filter utilities for raw video: medianfilter, scaling, and transitions. It also contains an encoder to MPEG-1/2 video streams from the YUV-format.

- **aenc**: Contains the source files for “mp2enc” which is MPEG-1 layer 2 audio compressor.²⁷
- **mplex**: Mplex is a simple two-stream audio/video multiplexer for MPEG-1/MPEG-2. It accepts an MPEG-1/2 video stream and/or an MPEG layer II/III audio stream and multiplexes them into a combined program/system stream according to the constraints specified. Many different types of output structure are supported along with presets for standard VCD and SVCD streams²⁸. Mplex is capable of automatically splitting the output stream into chunks of a specified size either independently or a sequence end/start points in the input video stream.
- **mjpeg, utils**: contains libraries to handle the new MJPEG movtar video format and a library to simplify MJPEG software and hardware playback.

B.5.11 MSYS toolkit

Package:	MSYS toolkit	Version:	1.0	Date:	1995-01-06
Exports:	encode, decode	Type:		License:	educational
Uses:		Url:	ftp://ftp.tek.com/tv/vnd/		
Author(s):	Guy Cherry, Tektronix				

The MSYS Toolkit is a set of C++ classes for building and manipulating MPEG system streams. These classes can be used to build MPEG system multiplexers and demultiplexers, and other MPEG related applications. The package is written to be instructional, and is therefore not optimised for efficiency.²⁹

B.5.12 mpeg2dec

Package:	mpeg2dec	Version:	0.3.1	Date:	Fri Dec 13 22:15:36 PST 2002
Exports:		Type:		License:	library/application
Uses:		Url:			GPL
Author(s):	Aaron Holtzman, Michel Lespinasse				

mpeg2dec is a test program for libmpeg2. It decodes mpeg-1 and mpeg-2 video streams, and also includes a demultiplexer for mpeg-1 and mpeg-2 program streams. It is purposely kept simple : it does not include features like reading files from a DVD, CSS, fullscreen output, navigation, etc... The main purpose of mpeg2dec is to have a simple test bed for libmpeg2.

B.5.13 mpeg2dec-mjpegtools

Package:	mpeg2dec	Version:	0.2.1-mjpegtools-0.7-1nr	Date:	24 Feb 2003
Exports:		Type:		License:	library/application
Uses:		Url:			GPL
Author(s):	Aaron Holtzman, Michel Lespinasse				

The jpegtools patch of *mpeg2dec* uses jpegtools to be able to produce VCD compliant MPEG1. The NR release of the rpm, makes a binary with a new name **mpeg2dec-mjpegtools** and without the libraries, so it can co-exist with other mpeg2dec versions.

²⁷It is not particularly good as encoders go, but is included for simplicity and completeness. It also has the virtue that for transcoding applications (e.g. AC3 to mp2) it can do sampling rate conversions. You need this to compress audio.

²⁸VCD and SVCD streams can be used in the *vcdimager* software to produce (S)VCD media.

²⁹The software is no longer available, as also noted on the web site <http://mpeg.org>.

B.5.14 smilutils

Package:	smilutils	Version:	0.3.0	Date:	02 Oct 2003
Exports:		Type:		tools	
Uses:		License:		GPL	
Url:	http://kino.schirmacher.de/article/view/70/1/7/				
Author(s):	Charles Yates				

smilutils features a bunch of new utilities for video fx, fullscreen DV playback (supporting all of kino's file formats and DV pipes) and even a utility to transcode DivX, mpeg and many other formats to Raw DV. Programs included: image2raw, ppm2raw, raw2dv, raw2image, raw2webcam, raw2yuv, rawmultiply, smil2raw, smil2wav, smil2yuv, tga2raw, and xwd2raw.

B.5.15 ImageMagick

Package:	ImageMagick	Version:	5.5.5	Date:	01 Mar 2003
Exports:		Type:		tools	
Uses:	bzip2, freetype, libjpeg, libpng, libtiff, libungif, zlib	License:		BSD-style	
Url:	http://www.imagemagick.org				
Author(s):	ImageMagick Studio				

ImageMagick is a collection of tools and libraries for converting displaying and editing pictures. It handles over 87 major formats.

B.6 Player software and presentation

B.6.1 Monkey Media

Package:	Monkey Media	Version:	0.5	Date:	December 20, 2002
Exports:		Type:		Media Library	
Uses:	gnome-vfs, gstreamer	License:		LGPL	
Url:	http://people.nl.linux.org/~jorn/monkey-media.html				
Author(s):	Jorn Baayen, Marco Pesenti Gritti, Bastien Nocera, Seth Nickell				

Monkey Media is a library for multimedia playback (of which only audio is supported at this point), offering support for seeking, managing volume, obtaining stream information like ID3 tags and bitrate, equalizer, and GStreamer visualization plugins.

B.6.2 Rhythmbox

Package:	Rhythmbox	Version:	0.4.1	Date:	December 20, 2002
Exports:		Type:		Player	
Uses:	gstreamer	License:		GPL	
Url:	http://www.rhythmbox.org/				
Author(s):	Jorn Baayen, Olivier Martin				

Rhythmbox takes its inspiration from Apple's iTunes application. The functionality includes importing audio CDs into mp3 or Ogg Vorbis format, play these music files and other music files, and burn new audio CDs from music files. Several visualisation plugins are available.

B.6.3 MPlayer

Package:	MPlayer	Version:	0.60	Date:	
Exports:		Type:		application	
Uses:	libmpeg2	License:		GPL	
Url:	http://www.mplayerhq.hu/				
Author(s):					

MPlayer is a movie player for Linux/Unix systems, e.g., based on X11. MPEG-1 and MPEG-2 formats are decoded with the native library libmpeg2. MPlayer supports a wide range of output drivers: X11, Xv, DGA, OpenGL, SVGAlib, fbdev, AAlib, SDL, VESA, and low-level drivers specific to hardware cards (Matrox, 3DFX and Radeon). Most of these drivers support software or hardware scaling for full-screen playback. MPlayer also support some MPEG decoder boards, such as DVB and DXR3/3 Hollywood+. An impressive list of supported software codecs is available from <http://www.mplayerhq.hu/DOCS/codecs-status.html>. Some codecs are decoded using modified win32 libraries.

B.6.4 XINE

Package:	Xine	Version:	0.9.20	Date:	April 2003
Exports:		Type:		License:	GPL
Uses:					
Url:	http://xine.sourceforge.net/				
Author(s):	Guenter Bartsch				

XINE plays MPEG-2 and MPEG-1 video, DVDs (unlocked/unencrypted only), video CDs, SVCDS, and AVI files (using proprietary Windows 32 codecs) with synchronised audio and video, and optionally full-screen using the Xv extensions in Xfree86 4.x. XINE has a modular, multi-threaded architecture for input, demuxer, decoder, and output plugins.

B.6.5 gxine

Package:	gxine	Version:	0.2	Date:	December 10, 2002
Exports:		Type:		License:	application
Uses:					GPL
Url:	http://xine.sourceforge.net/				
Author(s):	Guenter Bartsch				

gxine is both a GNOME frontend and a Mozilla plugin based on the core XINE library, xine-lib.

B.6.6 XMMS

Package:	XMMS	Version:	1.2.7	Date:	December 20, 2002
Exports:		Type:		License:	Multimedia player
Uses:					GPL
Url:	http://www.xmms.org				
Author(s):	Peter Alm				

XMMS is a multimedia player for Unix systems. XMMS stands for X Multimedia System and can play media files such as MP3, MOD's, WAV and others with the use of Input plugins.

B.6.7 XMovie

Package:	XMovie	Version:	1.9.7	Date:	December 10, 2002
Exports:		Type:		License:	application
Uses:					GPL
Url:	http://heroinewarrior.com/download.php3				
Author(s):	Heroine Virtual				

XMovie supports playback of MPEG-1, MPEG-2, MP3 audio, MP2 audio, AC3 audio and a range of other formats. According to the authors, "XMovie is primarily used for uncompressed, high resolution playback of output from an editing program". No statements about synchronised playback.

B.6.8 mpeg_play

Package:	mpeg_play	Version:	2.4	Date:	April 6, 1998
Exports:		Type:		Type:	application
Uses:		License:		License:	GPL
Url:	http://bmerc.berkeley.edu/frame/research/mpeg/mpeg_play.html				
Author(s):	Lawrence A. Rowe, Ketan Patel, Brian Smith				

The Berkeley MPEG Player, `mpeg_play`, is an MPEG player written in C. It uses X11 to display the decoded movies by default. It can optionally produce PPM files, SVGA graphics (Linux), Windows graphics calls, or work in a Mac window. It does not handle real-time synchronisation or audio streams.

B.6.9 Alsaplayer

Package:	Alsaplayer	Version:	0.99.73	Date:	December 2002
Exports:		Type:		Type:	Audio Player
Uses:		License:		License:	GPL
Url:	http://www.alsaplayer.org/				
Author(s):	Andy Lo-A-Foe				

AlsaPlayer is a PCM player. It is heavily multi-threaded and tries to exercise the ALSA library and driver. It has some very interesting features unique to Linux/Unix players. The goal is to create a fully pluggable framework for playback of all sorts of media with the focus on PCM audio data.

B.6.10 VideoLAN Client

Package:	VideoLAN Client	Version:	0.4.6	Date:	December 2002
Exports:		Type:		Type:	application
Uses:		License:		License:	GPL
Url:	http://www.videolan.org/vlc/				
Author(s):	Students from the École Centrale Paris				

The VideoLAN Client (VLC) is a MPEG player, that support MPEG-2 TS unicast or multicast and HTTP protocol over IPv4 and IPv6 networks. VLC also support slow motion, fast forward, pause, arbitrary jumps within the stream, DVD chapter selection, on-the-fly language and subtitles selection. It uses X11 to display decoded movies.

B.6.11 gqview

Package:	gqview	Version:	1.2.0	Date:	December 20, 2002
Exports:		Type:		Type:	Image browser
Uses:		License:		License:	GPL
Url:	http://gqview.sourceforge.net				
Author(s):	GQview team				

Image browser that features single click access to view images and move around the directory tree.

B.6.12 DirectPresenter

Package:	DirectPresenter	Version:		Date:	December 16, 2002
Exports:		Type:		Type:	ISO image generation
Uses:	DirectFB	License:		License:	GPL
Url:	http://team.gcu-squad.org/~fab/DP/				
Author(s):	Fab				

B.6.13 GImageView

Package:	GImageView	Version:	0.2.10	Date:	December 20, 2002
Exports:		Type:		Type:	Image browser
Uses:		License:		License:	GPL
Url:	http://gtkmmviewer.sourceforge.net/				
Author(s):	Takuro Ashie, Nyan2				

GImageView is a GTK+ based image viewer. It support tabbed browsing, thumbnail table views, directory tree views, drag and drop, reading thumbnail cache of other famous image viewers, and flexible user interface.

B.6.14 Freevo

Package:	Freevo	Version:	1.3.1	Date:	February 11, 2003
Exports:		Type:		Type:	digital video jukebox
Uses:		License:		License:	GPL
Url:	http://freevo.sourceforge.net/				
Author(s):	Krister Lagerström				

Freevo is an open-source digital video jukebox (PVR, DVR) based on Linux in addition to a number of open-source audio/video tools. MPlayer is used to play audio and video files.

B.6.15 MythTV

Package:	MythTV	Version:	0.8	Date:	March 16, 2003
Exports:		Type:		Type:	PVR
Uses:		License:		License:	GPL
Url:	http://www.mythtv.org/				
Author(s):	Isaac Richards				

MythTV is a open-source PVR. It supports TV watching, and recording to disk. There are also lots of plug-ins for listening to music, checking the weather, and playing games.

B.7 Streaming and other servers

B.7.1 Icecast

Package:	Icecast	Version:	1.3.12	Date:	2001
Exports:		Type:		Type:	Audio streamer
Uses:		License:		License:	GPL
Url:	http://www.icecast.org/				
Author(s):	Jack Moffitt, Barath Raghavan, Alexander Havang, Chad Armstrong and Jeremy Katz				

Icecast is an Internet audio broadcasting system based on MPEG audio technology. It implements to broadcast an audio stream to as many people as their bandwidth can support. The Icecast server supports clients as e.g., **shout**, WinAmp, or xmms.

B.7.2 VideoLAN Server

Package:	VideoLAN Server	Version:	0.4.4	Date:	11 Aug 2002
Exports:	MPEG transport stream	Type:		Type:	MPEG streamer
Uses:	libdvdread to play DVDs, libdvdcss for encrypted DVDs, libdvbpsi for DVB input	License:		License:	GPL
Url:	http://www.videolan.org/vls				
Author(s):					

VideoLAN is a project of students from the École Centrale Paris and other developers. The VideoLAN Server (vls) is designed for handling many MPEG sources and broadcasting data over

an IP network. VideoLan runs on GNU/Linux, Solaris, and Windows (port). The following MPEG sources are supported: MPEG1 PS, MPEG2 PS, MPEG2 TS, DVD, the Kfir video encoder and the WinTV-Nova satellite receiver. The output is MPEG2-TS over an IP network or to a file.

B.8 Metadata, search and retrieval

B.8.1 Advanced Authoring Format

Package:	Advanced Authoring Format	Version:	1.0.1	Date:	July 16, 2002
Exports:		Type:		License:	SDK
Uses:					
Url:	http://sourceforge.net/projects/aaf/				
Author(s):					

The file format **Advanced Authoring Format (AAF)** for use in professional multimedia creation and authoring comes with a cross-platform SDK. The AAF specification is administered by the AAF Association, which also coordinates development of the SDK.

B.9 Mastering

B.9.1 vcdimager

Package:	vcdimager	Version:	0.7 (unstable)	Date:	
Exports:		Type:		License:	application
Uses:					GPL
Url:	http://www.vcdimager.org				
Author(s):	Herbert Valerio Riedel				

GNU **VCDimager** is a mastering suite for authoring, disassembling and analyzing Video CD's and Super Video CD's. The core functionality consists of directly making Video CD BIN/CUE-style CD images from mpeg files, which (after being written to CDR(W) media) can be played on standalone VCD players or DVD players and on computers running GNU/Linux, MacOS, Win32 or any other OS capable of accessing VCD's. BIN/CUE images can be burned with the program **cdrdao** under GNU/Linux.

B.9.2 cdrdao

Package:	cdrdao	Version:	0.	Date:	2002-10-05
Exports:		Type:		License:	application
Uses:					GPL
Url:	http://cdrdao.sourceforge.net/				
Author(s):	Sourceforge				

Cdrdao records audio or data CD-Rs in disk-at-once (DAO) mode based on a textual description of the CD contents.

B.9.3 dvdauthor

Package:	dvdauthor	Version:	0.6.5	Date:	2003-11-05
Exports:		Type:		License:	application
Uses:					GPL
Url:	http://dvdauthor.sourceforge.net/				
Author(s):	Scott Smith				

dvdauthor is a set of tools for generating DVD files to be played back on a standalone DVD player. The use of the program is described in [7].

B.9.4 cdrtools

Package:	cdrtools	Version:	2.0	Date:	2002-12-26
Exports:		Type:		application	
Uses:		License:		GPL	
Url:	http://www.fokus.fhg.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html				
Author(s):	Jörg Schilling				

cdrtools is a set of tools for burning cd's. The toolset includes *mkisofs* and *cdrecord*.

B.9.5 cdrecord-prodvd

Package:	cdrecord-prodvd	Version:	2.0.0	Date:	2002-12-26
Exports:		Type:		application	
Uses:		License:		Proprietary	
Url:	http://www.fokus.fhg.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html				
Author(s):	Jörg Schilling				

cdrecord-prodvd is a version of *cdrecord* that can burn DVD's. It's not open source, but can be used for free for educational, research and private non-commercial purposes.

B.9.6 dvdrttools

Package:	dvdrttools	Version:	0.1.5	Date:	2003-07-12
Exports:		Type:		application	
Uses:		License:		GPLv2	
Url:	http://www.nongnu.org/dvdrttools/				
Author(s):	Bero - Savannah				

dvdrttools is a fork of *cdrtools*, with the primary goal of supporting writable DVD drives.

B.9.7 cdparanoia

Package:	cdparanoia	Version:	III-alpha9.8	Date:	March 2001
Exports:		Type:		application	
Uses:		License:		GPL	
Url:	http://xiph.org/paranoia/				
Author(s):	Monty				

Cdparanoia is a Compact Disc Digital Audio (CDDA) extraction tool. The application is built on top of the Paranoia library, which is doing the real work (the Paranoia source is included in the *cdparanoia* source distribution). Like the original *cdda2wav*, *cdparanoia* package reads audio from the CDROM directly as data, with no analog step between, and writes the data to a file or pipe in WAV, AIFC or raw 16 bit linear PCM.

B.9.8 mpg32ogg

Package:	mpg32ogg	Version:	0.11	Date:	December 2002
Exports:		Type:		application	
Uses:		License:		Artistic	
Url:	http://faceprint.com/code/				
Author(s):	Nathan Walp				

mpg32ogg is a *Perl* script to convert MP3 files to Ogg Vorbis files, retaining ID3 information, bitrate, and optionally renaming the output files, as well as deleting the originals.

B.10 Digital TV software

We only list the packages that are of a more general nature. We do not list packages that need the LinuxTV kernel (see linuxtv.org), or need special DVB cards. More information in digital TV software can be found in a separate report [15].

B.10.1 DVBstream

Package:	dvbstream	Version:	0.4pre2	Date:	2002-01-31
Exports:		Type:		library,application	
Uses:	ts-rtp package from LinuxTV			License:	GPL
Url:	http://linuxstb.org/dvbstream/				
Author(s):	David Podeur, Dave Chapman				

DVBstream is based on the ts-rtp package available at <http://www.linuxtv.org>. It broadcasts a (subset of a) DVB transport stream over a LAN using the rtp protocol. There were a couple of small bugs in the original ts-rtp application, which have been fixed within this project. This software requires Linux, a supported DVB card, the Linux DVB drivers from www.linuxtv.org and a kernel with Multicast networking enabled.

B.10.2 librtsp

Package:	librtsp	Version:	0.1-1	Date:	2001
Exports:		Type:		library	
Uses:		License:		GPL	
Url:	http://librtsp.sourceforge.net				
Author(s):	Roland Dreier				

Librtsp is a library for writing RTP/RTCP applications. The code was originally part of gnome-o-phone (see <http://gphone.sourceforge.net>).

B.10.3 HIØ DVB package

Package:	HIØ DVB package	Version:		Date:	2001-10-31
Exports:		Type:		collection	
Uses:	Mplayer 0.50, bfr 1.3, dvbstream 0.3, mpegtools			License:	GPL
Url:	nolongeravailable.				
Author(s):	Håvard Rast Blok, Høgskolen i Østfold				

This package contains the programs necessary for receiving the RTP stream of high quality, MPEG2 video from Høgskolen i Østfold.³⁰ The package includes the following programs and libraries:

- LinuxTV (mpegtools and DVB driver) <http://www.linuxtv.org/>
- mpegtools, see Section B.5.8.
- DVBStream (dumprtp) version 0.3 from <http://www.linuxstb.org/>. See Section B.10.1. dvbstream is used to broadcast a DVB transport stream over a LAN using the RTP protocol.
- The buffering software **bfr 1.3** by Mark Glines from <http://www.glines.org/software/buffer.html>. The software buffers data in order to avoid artefacts from jitter and delays on the network.
- OST (some header files only, probably necessary for a stand-alone version.)
- MPlayer 0.50.

³⁰The HIØ DVB package can be seen as a small add-on distribution for digital TV software to Linux. The purpose of this distribution is to turn a Linux computer into a set-top box.

B.11 IP phones, multimedia phones and phone conference equipment

B.11.1 Gnome-o-Phone

Package:	Gnome-o-Phone	Version:	0.5.2	Date:	
Exports:		Type:		Internet Telephone	
Uses:	librtsp, libgsm, GNOME	License:		GPL	
Url:	http://gphone.sourceforge.net/				
Author(s):	Roland Dreier, Rob Johnson				

Gnome-o-Phone (gphone for short) is an Internet telephone. Calls are made over the Internet without using the PSTN infrastructure. gphone uses RTP/RTCP, and gphone complies to the standard to be able to talk to *speakfreely*. Gphone requires the GSM compression library.

B.11.2 Linphone

Package:	Linphone	Version:	0.9.0	Date:	
Exports:		Type:			
Uses:		License:		GPL	
Url:	http://www.linphone.org/				
Author(s):	Simon Morlat				

Linphone is a web phone for the Gnome Desktop and Linux (can also be used under KDE). It works with a simple user interface. Since version 0.9.0, linphone can be compiled and used without gnome, in console mode, by using the program called *linphonec*. Linphone includes a variety of codecs. The Speex codec allows high quality talks even with slow internet connections, like 28k modems. Linphone understands the SIP protocol.

B.11.3 RAT

Package:	RAT	Version:	4.2.24	Date:	
Exports:		Type:			
Uses:		License:		publicly available	
Url:	http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/index.html				
Author(s):	Vicky Hardman, Peter Kirstein, Angela Sasse et.al.				

RAT (Robust Audio Tool) is a an open-source audio conferencing and streaming application that allows users to participate in audio conferences over the Internet. These can be between two participants directly, or between a group of participants on a common multicast group.

RAT requires no special features for point-to-point communication, just a network connection and a soundcard. For multiparty conferencing RAT uses IP multicast. RAT is based on IETF standards, using the RTP profile for audio and video conference with minimal control. RAT features a range of different rate and quality codecs, receiver based loss concealment to mask packet losses, and sender based channel coding in the form of redundant audio transmission. RAT runs on a range of platforms: FreeBSD, HP-UX, IRIX, Linux, NetBSD, Solaris, SunOS, and Windows 95/NT.

B.12 Video conference

B.12.1 GnomeMeeting

Package:	GnomeMeeting	Version:	1.5.2	Date:	2003-09-11
Exports:		Type:		Video conference	
Uses:	OpenH323, PWlib	License:		GPL	
Url:	http://www.gnomemeeting.org/				
Author(s):	Damien Sandras				

GnomeMeeting is a GNU/Linux H.323 client, which is compatible with Netmeeting and other H.323 products. It has been designed for the Gnome desktop, but it also works with other environments.

GnomeMeeting is H.323 compliant. It supports H.245 Tunneling, Fast Start, auto answering of incoming calls, and do not disturb mode. It GnomeMeeting supports the following audio codecs: LPC10, GSM-06.10, MS-GSM, G.711-Alaw, G.711-uLaw, G.726; and video codecs: H.261-QCIF and H.261-CIF. Other features include Quicknet hardware devices, the G.732.1 audio codec, `callto://` URLs, gatekeeper support, video bandwidth control and text chat.

B.12.2 vic

Package:	vic	Version:	2.8	Date:	96-07-22
Exports:		Type:		License:	Video conference
Uses:		License:			
Url:	http://www-nrg.ee.lbl.gov/vic/				
Author(s):	Steven McCanne, Van Jacobson				

The UCP/LBNL video tool, **vic**, is a real-time, multimedia application for video conferencing over the Internet. It is based on RTP. The audio tool **vat** and the whiteboard tool **wb** can be used in addition. Other tools include the session directory tool **sdr**, the multimedia conference control **mmcc**, the Xerox PARC Network Video tool **nv**, and the INRIA Video conferencing System **ivs**.

The features of **vic** include an Intra-H.261 video encoder, voice switched viewing windows, multiple dithering algorithms, interactive title generation, and routing of decoded video to external video ports.

B.13 Text To Speech and Speech to text

B.13.1 Festival

Package:	Festival	Version:	1.4.2	Date:	
Exports:		Type:		License:	Speech syntezis
Uses:		License:			GPL?
Url:	http://www.cstr.ed.ac.uk/projects/festival/				
Author(s):	Alan W Black (CMU), Rob Clark (CSTR), Richard Caley (CSTR), and Paul Taylor				

Festival is a general multi-lingual speech synthesis system developed at CSTR. It offers a full text to speech system with various APIs, as well an environment for development and research of speech synthesis techniques. It is written in C++ with a Scheme-based command interpreter for general control.

B.13.2 Xvoice

Package:	Xvoice	Version:	0.9.5	Date:	
Exports:		Type:		License:	Speech to text
Uses:	ViaVoice	License:			GPL
Url:	http://www.compapp.dcu.ie/~tdoris/Xvoice/				
Author(s):	Brian Craft				

Xvoice Xvoice enables continuous speech dictation and speech control of X applications. It uses the IBM ViaVoice speech recognition engine to convert users' speech into text. When in dictation mode Xvoice passes this text directly to the currently focused X application. In command mode, Xvoice matches the speech with predefined, user-modifiable, key sequences or commands.

ViaVoice for Linux must be downloaded separately from IBM (www.ibm.com).

C Other Multimedia-oriented distributions

We found other multimedia related distributions and distribution projects, and present some of these initiatives. Besides these distributions, there are Linux-oriented software packages available for special purposes. This includes the software distributed for specific set-top boxes, e.g., the Nokia Mediaterminal (See e.g., [15]).

C.1 AGNULA

AGNULA (acronym for A GNU/Linux Audio distribution, pronounced with a strong g) is the name of a project funded by the European Commission (number of contract: IST-2001-34879; key action IV.3.3, Free Software: towards the critical mass). The project is coordinated by the Centro Tempo Reale in Firenze (<http://www.centrotemporeale.it>) and involves several research centres and institutions:

- IRCAM (Institut de Recherche et Coordination Acoustique/Musique), Paris (<http://www.ircam.fr/>);
- Universitat Pompeu Fabra, Music Technology Group, Barcelona (<http://www.iaa.upf.es/mtg/>);
- Kungl. Tekniska Högskolan (KTH), Music Acoustics Group, Stockholm (<http://www.speech.kth.se/music/>);
- Free Software Foundation Europe (<http://fsfeurope.org/>);
- and Red Hat France (<http://www.redhat.fr/>).

AGNULA supports the development of derived distributions of Debian GNU/Linux and Red Hat GNU/Linux. Dedicated to audio and multimedia these distributions will be called DeMuDi (after Debian Multimedia Distribution) and ReHMuDi (after Red Hat Multimedia Distribution) with the following features: easy to use (auto-installing, with auto-configuration, plug-and-play), full-featured GNU/Linux system (completely configurable and customisable, reliable and secure, strongly linked to networking the Internet, and easy to upgrade); as in all Debian and in most Red Hat distributions, most of the software is released with GNU/GPL licence, and all included software will be open source software.

The deliverables and reports of the AGNULA project are available at <http://www.agnula.org/project/deliverables>

The packages included in AGNULA, according to <http://www.agnula.org/user/packages>, are grouped into the following categories:

- Audiodevel
- CD-tools
- Compression
- Conversion
- Digital DJ
- DSP
- MIDI
- ModTrack
- Multitrack
- Network
- Notation
- Play/Record
- Plugin
- Sounddriver
- Soundedit
- Synthesis
- Video
- Visual

C.2 DeMuDi: Debian Multimedia Distribution

DeMuDi is part of the AGNULA project. The focus is on building packages for the Debian distribution. The packages included in Demudi are grouped into the following categories [8]:

- Sound Editors
- Multitrackers
- Software Synthesis
- MIDI and Trackers
- Compression and Internet
- CD Production
- Miscellaneous Tools
- Notation
- Visualisation
- Speech
- Video and Graphics
- Development Tools

C.3 Planet CCRMA at Home

Planet CCRMA at Home is a collection of RPMS that can be added to a computer running Red Hat 7.2, 7.3 or 8.0 to transform it into an audio workstation. The features include a low-latency kernel, current ALSA audio drivers and a set of applications for music, MIDI and audio. The project was started by Juan Reyes and Fernando Lopez-Lezcano at the Stanford University Center for Computer Research in Music and Acoustics (CCRMA).

C.4 NIST DASE

ATSC DASE, the American Digital Interactive Television standard, have released a reference implementation of their set-top box which runs on Linux. The ATSC DASE specification is quite similar to the MHP specification for digital TV (see www.dvb.org). It includes interfaces for JavaTV, HAVi and DAVIC. The reference implementation includes full source code, a JavaTV implementation, and a HAVi implementation. See <http://www.cmr.nist.gov/dase/> for more information. We refer to our report on digital TV [15].

C.5 MoviX

The MoviX [5] project is a series of three tiny Linux CD distributions containing all the software to boot from a CD and play multimedia files through MPlayer. Using the distribution does not change anything with the installation on the machine. The MoviX project can be found at <http://movix.sf.net>. The reason for having such a distribution is twofold:

- By inserting the CD and rebooting the PC the PC can be used as a general purpose multimedia player, independent of the operating system installed before, e.g., for playing audio or video files. This is the purpose of *MoviX* and *MoviX²*. While *MoviX* is a distribution for a multimedia player, the distribution Knoppix (see <http://www.knopper.net/knoppix/>) uses the same paradigm as a more general purpose distribution.
- By attaching the *eMoviX* kernel (size ca. 8MB) to multimedia files the resulting CD or DVD can be used by booting the PC without any other software installed, i.e. the multimedia files and playing software are on the same medium. (The multimedia files still can be played on other players and operating systems.) The production of a *eMoviX* CD can be done on both Windows and Linux systems.

Part III

Details on mediAkit

In the following we show some details and examples on how to use and prepare software packages for the **mediAkit** distribution.

D Details on the DVD Use Case

D.1 Creating a VCD

To produce a VCD we transcode the original mpg-file into a VCD compliant MPEG1. For this a version of `mpeg2dec` patched by the `mjpegtools` project must be installed. For processing the video-stream the following command-line was used:

```
mpeg2dec -s -o YUV original.mpg | yuvscaler -O VCD
  | mpeg2enc -f 1 -r 16 -o video_vcd.m1v
```

The audio stream must be encoded with the mp2 codec: Dump the audio to a raw pcm-file using `mplayer`, and encode into a mp2 audio stream:³¹

```
mplayer -vc null -vo null -ao pcm -aofile audio.pcm original.mpg
cat audio.pcm | mp2enc -V -v 2 -o audio_vcd.mp2
```

Now we multiplex the audio stream and the video stream into a single stream using `mplex`. If the A/V should be out of synch use `-o SYNCH_VALUE`.

```
mplex -f 1 audio_vcd.mp2 video_vcd.m1v -o vcd_stream.mpg
```

Now create the VCD image with `vcdimager`, and burn the VCD using `cdrdao`:

```
vcdimager -v -l LABEL vcd_stream.mpg
cdrdao write --device 0,6,0 --driver generic_mmc vcd_stream.cue
```

A list of possible devices can be found by using `cdrecord -scanbus`. A list of supported drivers can be found at <http://cdrdao.sourceforge.net/drives.html>.

D.2 Editing the mpeg-file

The second task was editing the mpeg file, for which we chose the application `kino`. However, `kino` only works on raw video, so we had to transcode the mpeg-stream into a DV-encoded stream. We first tried to do this using `transcode`:

```
transcode -x mpeg2 -y dv -Z 720x480 -i infile.mpg -o outfile.avi
```

The resulting stream was flipped upside down, and some colors were wrong. To fix this we added the parameters `-k` and `-z`:

```
transcode -x mpeg2 -y dv -Z 720x480 -k -z -i infile.mpg -o outfile.avi
```

We also tried using `mencoder` that comes with `mplayer`:

```
mencoder infile.mpg -ovc libdv -oac pcm -vop scale=720:480 -o outfile.avi
```

The resulting stream had the correct orientation, and the colors were approximately right. Note that we had to change the size of the stream from 352×288 to 720×480 .

³¹Alternatively the applications `toolame` and `ac3dec` (which is obsoleted by `a52dec`) could be used for this purpose.

D.3 Creating a DVD-compatible mpeg-file

In order to be burned on a DVD the original file has to be converted into a DVD compatible file. We convert the video into a raw format that we can scale, scale this and encode it using MPEG-2:³²:

```
mpeg2dec -o YUV original.mpg | yuvscaler -O SIZE_720x576 | yuvdenoise |
mpeg2enc -f 8 -F 3 -o video.m2v
```

Alternatively, *mencoder* or *transcode* can be used to decode the original mpeg-movie into an yuv-file.

The sound is coded like for the VCD case, but the sampling rate must be 48000Hz:

```
mplayer -vc null -vo null -ao pcm -aofile audio.pcm original.mpg
cat audio.pcm | mp2enc -s -r 48000 -o audio.mp2
```

Now the video and audio can be multiplexed into the DVD-compatible mpeg-file:

```
mplex -f 8 audio.mp2 video.m2v -o dvd_stream.mpg
```

D.4 Converting a sequence of pictures into a DVD-compatible mpeg-file

To convert a sequence of gif-images into a mpeg2-video that can be added to a DVD we convert the pictures into jpeg, and then use the application *jpeg2yuv* to convert the pictures into a yuv-movie. Note that *jpeg2yuv* has a bug that make the program crash if the pictures are not in the correct size of 720×576 . Therefore it is necessary to resize the images before converting, which can be done by resizing or adding borders. (see *jpeg2yuv -help*).

To convert from gif to jpeg the program *convert* from the ImageMagick package is used. Notice that the ratio between height and width is not kept. To avoid this, e.g., the *-border* option can be used.

```
convert image0000.gif -resize 720x576! image0000.jpg
```

The sequence image0000.jpg, image0001.jpg, ... can be transferred into a yuv-encoded video, and then coded into mpeg-2:

```
jpeg2yuv -Ip -f 25 -j image\%04d.jpeg > video.yuv
cat video.yuv | yuvdenoise | mpeg2enc -f 8 -o video.m2v
```

D.5 Editing and adding sound

The program *audacity* can be used to edit sound, and mix sound from different file sources. The sound can then be saved in wav-format. These files can be encoded in mpeg-2:

```
mp2enc -r 48000 -s -o audio.mp2 < audio.wav
```

This audio can then be multiplexed with a mpeg2-encoded video-file as described in Section D.3.

D.6 Creating a DVD

To make a DVD-image you need the *dvdauthor* program. The first step is to make the directory structure for the DVD. This is done with the following command:

```
dvddirgen -o /tmp/dvd
```

A directory will be created on */tmp/dvd*. Now the mpeg-files have to be added using the following command:

```
dvdauthor -o /tmp/dvd 1.mpg 2.mpg 3.mpg ...
```

³²For encoding into MPEG-2 we need the patched version of *mpeg2dec* that we used to create the VCD.

One chapter of the DVD will be used for each mpg-file, but this can be overridden using the *-chapter* option. After all the files are added we can create the table of contents:

```
dvdauthor -T -o /tmp/dvd
```

We are now ready to make an iso-image and burn the DVD:

```
mkisofs -dvd-video -o /tmp/dvdimage /tmp/dvd  
dvdrecord -dao -eject dev=1,0,0 /tmp/dvdimage
```

In our case *dvdrecord* did not work, which is a known error. A patch is available, but the patch wasn't updated for the latest version of *dvdrecord*. Therefore we used *cdrecord-prodvd* instead. The produced DVD worked in an ordinary DVD-player.

```
cdrecord-prodvd -v dev=1,0,0 -eject /tmp/dvdimage
```

E Software Packaging

Media Folder is a simple, yet unfinished library written by one of the authors. It will be used as an example to demonstrate some steps in the software packaging process. The package is available from <http://www.nr.no/~oka/software/mfolder/>. For another example, see the GNU *hello* package from the FSF (<http://www.gnu.org>).

E.1 Package Layout

- **AUTHORS** - text file that describes how to reach the package authors
- **autogen.sh** - script to regenerate autoconf and automake setup.
- **ChangeLog** - text file for recording changes and modifications to the package
- **configure.ac** - autoconf-specific rules from which autoconf generates a platform-neutral configure script
- **COPYING** - text file with licensing terms of the software package
- **HACKING** - special development policies and conventions
- **INSTALL** - text file that describes how to install the software
- **Makefile.am** - *automake*-specific rules from which a standards-compliant **Makefile.in** is generated
- **mfolder.pc.in** - pkgconfig³³ configuration file to be processed by *autoconf*-specific variables with versioning and link details for libraries.
- **mfolder.spec.in** - RPM specification file to be processed by *autoconf* for RPM package building
- **NEWS** - a list of new features and modified functionality
- **README** - textual description of the software package.
- **src/** - the directory that contains the source code
- **src/Makefile.am** - *automake*-specific rules from which a standards-compliant **Makefile.in** is generated
- **src/mfolder.h** - C header file with function prototype and variable definitions
- **src/mfolder-types.h** - C header file with function prototype and variable definitions
- **src/mfolder.c** - C source file that the library is built from.
- **src/mfolder-tests.c** - C source file for test executables using the library
- **THANKS** - text file with a list of persons who contributed/helped.
- **TODO** - text file with description of planned improvements.

E.1.1 Makefile.am

Makefile.am in the top package directory contains *automake* specific build rules. It is possible to specify sub-directories, special directories on the installed system with a certain prefix where files (**mfolder.pc**) will be installed, and extra files that should go with the package (such as **mfolder.spec.in**, and the autoconf-generated **mfolder.spec** with the **Version:** field set to the to-be-released version of the package.

`$(libdir)/pkgconfig` is equal to `/usr/lib/pkgconfig`, when the source package is configured with the autoconf prefix `/usr`.

We also specify the **RPMFLAGS** variable and a traditional *make* target for building RPM (A.2.2) packages of the package.

```
SUBDIRS = src
pkgconfigdir = $(libdir)/pkgconfig
pkgconfig_DATA = mfolder.pc
EXTRA_DIST = HACKING mfolder.spec.in mfolder.pc.in mfolder.spec

RPMFLAGS=      --define ‘‘_sourcedir $$PWD’’ \
```

³³ *pkgconfig* is a program and *autoconf* macro to get compile and link flags for libraries the programs are linked with. It requires the library package to install a description file with the needed information.

```

--define ‘_builddir $$PWD/BUILD’ \
--define ‘_rpmdir $$PWD’ \
--define ‘_srcrpmdir $$PWD’
rpm: dist
    -$(RM) -r BUILD
    mkdir BUILD
    rpmbuild $(RPMFLAGS) -ba --clean mfolder.spec

```

Notice the rpm target which builds a SRPM and the binary RPM packages from the *autoconf*-generated `mfolder.spec` specification.

E.1.2 mfolder.pc.in

Below is a *pkgconfig*-configuration file to be replaced by a *autoconf*-generated version. The file contains versioning and link details for libraries. The content of the variables `MFOLDER_PKG_DEPS` and `VERSION` are expanded by *configure* and eventually stored in a file named `mfolder.pc`.

```

prefix=@prefix@
exec_prefix=@exec_prefix@
libdir=@libdir@
includedir=@includedir@

Name: mfolder
Description: Media Folder library
Requires: @MFOLDER_PKG_DEPS@
Version: @VERSION@
Libs: -L${libdir} -lmfolder
Cflags: -I${includedir}/mfolder-@VERSION@

```

E.1.3 mfolder.spec.in

Below is a RPM specification file to be processed by *autoconf* in order to make it easy to build new RPM packages for future releases of the package. The content of the `VERSION` variable is expanded by *configure* and stored in a new file, `mfolder.spec`.

```

%define ver @VERSION@
%define rel 1
%define prefix /usr

Summary: The Media Folder library
Name: mfolder
Version: %ver
Release: %rel
URL: http://www.nr.no/~oka/software/mfolder/
Source0: %{name}-%{version}.tar.gz
License: LGPL
Group: Development/Libraries
BuildRoot: %{_tmppath}/%{name}-root
Prereq: /sbin/ldconfig
Requires: gnome-vfs2 >= 2.0, glib2 >= 2.0
BuildRequires: gnome-vfs2-devel >= 2.0, glib2-devel >= 2.0
Prefix: %prefix

%description
Media Folder is a library for accessing and storing media files
in default locations.

%package devel
Summary: The static libraries and header files for Media Folder library
Group: Development/Libraries

```

```
Requires: %{name} = %{version}

%description devel
The mfolder-devel package contains the static libraries and header
files for the Media Folder library. For developing
programs that use Media Folder, mfolder-devel must be installed.

%prep
%setup -q

%build
./configure --prefix=%prefix
if [ '$SMP' != '' ]; then
    (make 'MAKE=make -k -j $SMP'; exit 0)
    make
else
    make
fi

%install
rm -rf $RPM_BUILD_ROOT

%makeinstall

%post
/sbin/ldconfig

%postun
/sbin/ldconfig

%clean
rm -rf $RPM_BUILD_ROOT

%files
%defattr(-,root,root)
%doc README AUTHORS COPYING NEWS THANKS TODO ChangeLog
%{prefix}/lib/*.so.*

%files devel
%defattr(-,root,root)
%{_libdir}/*.a
%{_libdir}/*.la
%{_libdir}/*.so
%{_libdir}/pkgconfig/mfolder.pc
%{_includedir}/mfolder-%{ver}/*.h

%changelog
* Tue Nov 12 2002 Ole Aamot <oka@nr.no>
- Initial build.
```

E.1.4 configure.ac

A platform-independent configure script is expanded by *autoconf* from this template:

```
AC_INIT(src/mfolder.c)
AC_CANONICAL_TARGET([])
AM_INIT_AUTOMAKE(mfolder, 0.1.0)
AM_MAINTAINER_MODE
AM_CONFIG_HEADER(config.h)
AC_CONFIG_SRCDIR([src/mfolder.c])
```

```
# Checks for essential programs.
AC_PROG_CC
AC_PROG_CC_STDC
AC_ISC_POSIX
AC_HEADER_STDC
AM_PROG_LIBTOOL

# Check for essential libraries.
PKG_CHECK_MODULES(MFOLDER,
    gnome-vfs-2.0 >= 2.0 \
    glib-2.0 >= 2.0)
MFOLDER_PKG_DEPS='glib-2.0, gnome-vfs-2.0'

# Checks for header files.
AC_HEADER_STDC
AC_CHECK_HEADERS([stdlib.h string.h errno.h sys/stat.h])

# Checks for typedefs, structures, and compiler characteristics.
AC_C_CONST

# Checks for library functions.
AC_CHECK_LIB(socket, connect)

# Modify the pkgconfig path.
AC_ARG_WITH(pkg-config-path,
    AC_HELP_STRING([--with-pkg-config-path],
        [colon-separated list of pkg-config(1) dirs]),
    [export PKG_CONFIG_PATH=${withval}])

AC_SUBST(MFOLDER_LIBS)
AC_SUBST(MFOLDER_CFLAGS)
AC_SUBST(MFOLDER_PKG_DEPS)

AC_OUTPUT([
Makefile
mfolder.spec
mfolder.pc
src/Makefile
])
```

E.1.5 src/Makefile.am

```
CFLAGS= -g -Wall $(MFOLDER_CFLAGS) $(MFOLDER_CFLAGS) -DDEBUG_MFOLDER
LDFLAGS= $(MFOLDER_LIBS) $(MFOLDER_LIBS)

lib_LTLIBRARIES = libmfolder.la

libmfolder_la_SOURCES = mfolder.c
libmfolder_la_LDFLAGS = -version-info 0:1:0
libmfolder_la_LIBADD = # @INTLLIBS@

libmfolderincdir = $(includedir)/mfolder-@VERSION@
libmfolderinc_HEADERS = mfolder.h mfolder-types.h

DEPS = libmfolder.la
LDADDS = libmfolder.la

noinst_PROGRAMS = mfolder-tests mfolder-tests-dbg

# Build from mfolder-tests.c and libmfolder.la
```

```
mfolder_tests_SOURCES = mfolder-tests.c
mfolder_tests_LDADD = libmfolder.la # @INTELLIBS@

# Create an easier-to-debug version.
mfolder_tests_dbg_SOURCES = mfolder-tests.c
mfolder_tests_dbg_LDADD = libmfolder.la
mfolder_tests_dbg_LDFLAGS = -static $(LDFLAGS)
```

E.1.6 autogen.sh

The `autogen.sh` script contains

```
#!/bin/sh
autoheader
aclocal
automake
autoconf
```

If `configure.ac` or `Makefile.am` needs to be modified, then the system needs at least *autoconf* 1.6.3, *automake* 2.53, and *libtool* 1.4.2 in order to rebuild `configure` and `Makefile.in` with the `autogen.sh` script.

```
$ wget http://www.nr.no/~oka/software/mfolder/mfolder-0.1.0.tar.gz
$ tar zxvf mfolder-0.1.0.tar.gz
$ cd mfolder-0.1.0/
$ ./autogen.sh
```

E.2 Build Instructions

The following instructions are necessary to build the package. The machine in this example is running *rpm* 4.1, *rpm-build* 4.1, *wget* 1.8.2, and *tar* 1.13.25.

```
$ wget http://www.nr.no/~oka/software/mfolder/mfolder-0.1.0.tar.gz
$ tar zxvf mfolder-0.1.0.tar.gz
$ cd mfolder-0.1.0/
$ ./configure
$ make rpm
```

The `rpm` target generates a Source RPM in the current directory, and two Binary RPM packages in the `i386/` subdirectory:

```
mfolder-0.1.0-1.src.rpm
i386/mfolder-0.1.0-1.i386.rpm
i386/mfolder-devel-0.1.0-1.i386.rpm
```

E.3 Install Instructions

To install the RPM packages on a compatible system, root permissions is required.

```
$ su
[enter root password]
# rpm -Uvh i386/mfolder-0.1.0-1.i386.rpm
# rpm -Uvh i386/mfolder-devel-0.1.0-1.i386.rpm
```

F Software Patches

When developers write enhancements or fixes to the original source package and would like to see the modifications included in the next version, it is common practice to use the *diff* program. Given the correct parameters *diff* creates a file that contains all of the modifications between two source files, the old without the modifications and the new file with the modifications according to special patch formatting rules.

F.1 Making a patch using *diff*

Although it is possible to use the *diff* program for only two files, it is useful to keep two separate directories:

- the original source tree
- the modified source tree

We made some modifications to `filmgimp.spec.in`, from which the RPM (see Section A.2.2) build specification is generated by *autoconf* for the `filmgimp` (B.3.4) package.

```
$ emacs filmgimp-0.11/filmgimp.spec.in
```

After the file (`filmgimp-0.11/filmgimp.spec.in`) was modified, we moved the original directory into a separate directory (`filmgimp-0.11.mod/`):

```
$ mv filmgimp-0.11/ filmgimp-0.11.mod/
```

Then the original source archive was uncompressed

```
$ tar zxvf filmgimp-0.11.tar.gz
```

At last *diff* was run and its output was compressed

```
$ diff --unified --recursive --new-file \
  filmgimp-0.11 filmgimp-0.11.mod/ >patch-spec-2002-12-18
$ gzip -9 patch-spec-2002-12-18
```

The steps produces a compressed patch file called `patch-spec-2002-12-18.gz`.

Example: `patch-spec-2002-12-18`

The content of `patch-spec-2002-12-18` is

```
--- filmgimp-0.11/filmgimp.spec.in      2002-12-04 08:10:08.000000000 +0100
+++ filmgimp-0.11.build/filmgimp.spec.in  2002-12-19 11:38:29.000000000 +0100
@@ -7,7 +7,12 @@
 Source0: http://prdownloads.sourceforge.net/filmgimp/filmgimp-@GIMP_VERSION@.tar.gz
 URL: http://filmgimp.sourceforge.net
 BuildRoot: %{_tmppath}/%{name}-root
-#Requires: /sbin/ldconfig FIXME: Find dependencies
+BuildRequires: gtk+-devel >= 1.2.10, glib-devel >= 1.2.10
+BuildRequires: libtiff-devel >= 3.5.7, libjpeg-devel >= 6b
+BuildRequires: libpng-devel >= 1.2.2
+Requires: gtk+ >= 1.2.10, glib >= 1.2.10,
+Requires: libtiff >= 3.5.7, libjpeg >= 6b, libpng >= 1.2.2
+PreReq: /sbin/ldconfig, grep, fileutils
 Prefix:  %{_prefix}
```

```
%description
```

It is common and usually also expected that a `ChangeLog` entry such as

```
2002-12-19 Ole Aamot <oka@naos.nr.no>
```

```
* filmgimp.spec.in: Add BuildRequires: and Requires:
```

is submitted by mail with the patch file to the package maintainer team. It is not considered good practise to send a patch that contains direct modifications against the `ChangeLog` file.

F.2 Applying a patch using *patch*

When the developer team receives a new patch file created by *diff*, as above, the developers responsible for the package maintenance, will consider if the modifications are correct, and eventually reject or accept the patch into the official source tree by using the *patch* tool. To apply the patch file with *patch* in the top level directory:

```
$ cd filmgimp-0.11/  
$ gzip -d < ../patch-spec-2002-12-18.gz | patch -p1  
patching file filmgimp.spec.in
```


G List of packages in the mediakit distribution

G.1 Programs

- audacity-1.0.0
- cdrdao-1.1.7
- cdrecord-2.0
- dvdauthor-0.5.0
- dvdrecord-0.1.2
- ImageMagick-5.4.7
- kino-0.6.4
- mediakit-0.3
- mediakit_dvd-0.1
- mkisofs-2.0
- mplayer-0.90rc4
- vcdimager-0.7.12
- xine-0.9.19

G.2 Libraries

- aalib-1.4rc5
- alsa-lib-0.9.1
- audiofile-0.2.3
- esound-0.2.28
- flac-1.1.0
- glut-3.7
- lame-3.93.1
- libdv-0.99
- libdvdcss-1.2.6
- libfame-0.9.0
- libjpeg-6b
- libmng-1.0.4
- libogg-1.0
- libpng-1.2.2
- libquicktime-0.9
- libtiff-3.5.7
- libungif-4.1.0
- libvorbis-1.0
- lirc-0.6.5
- mjpegtools-1.6.1
- mpeg2dec-0.3.1
- mpeg2dec-0.2.1-mjpegtools-0.7
- SDL-1.2.4
- wxGTK-2.2.9
- xvidcore-0.9.0

H NR mediAkit on Debian

Updated 2003-08-04 14:22.

The initial **mediAkit** release was developed and tested on RedHat 8.0. We also target Debian 3.0 (“woody”). For more information whether packages are available on Debian see <http://packages.debian.org/>.

H.1 Packages not yet available in Debian

The following packages are not available on Debian, by various reasons.

- mediakit_gui 0.1.0
- mediakit_dvd 0.1.0
- dvdauthor 0.5.3, see <http://packages.debian.org/dvdauthor>
- dvdrecord 0.1.2, see <http://packages.debian.org/dvdrecord>
- mjpegtools 1.6.1
- mplayer 0.90, see <http://packages.debian.org/mplayer>
- xine 0.9.19, see <http://packages.debian.org/xine>
- xsox 0.78, see <http://packages.debian.org/xsox>
- mpeg2dec-0.2.1-mjpegtools 0.7

H.2 Packages available in latest stable release of Debian (“woody”)

- libdvdread2 0.9.2-0.1, see <http://packages.debian.org/stable/libs/libdvdread.html>
- imagemagick 4:5.4.4.5-1woody1, <http://packages.debian.org/stable/graphics/imagemagick.html>
- audacity 0.98-3, <http://packages.debian.org/stable/sound/audacity.html>
- cdrecord 4:1.10-7, <http://packages.debian.org/stable/otherosfs/cdrecord.html>
- glame 0.6.2-2, <http://packages.debian.org/stable/sound/glame.html>
- kino 0.50-3, <http://packages.debian.org/stable/graphics/kino.html>
- soundtracker 0.6.4-0.3, <http://packages.debian.org/stable/sound/soundtracker.html>
- sweep 0.1.1-3, <http://packages.debian.org/stable/sound/sweep.html>

H.3 Packages available in latest testing release of Debian (“sarge”)

- cdrdao 1:1.1.7-0.1, <http://packages.debian.org/testing/otherosfs/cdrdao.html>
- vcdimager 0.7.14-1, <http://packages.debian.org/testing/otherosfs/vcdimager.html>
- mpeg2dec 0.3.1-3, <http://packages.debian.org/testing/x11/mpeg2dec.html>

H.4 Packages available in latest unstable release of Debian (“sid”)

- spiralsynthmodular 0.2.1-2, <http://packages.debian.org/unstable/sound/spiralsynthmodular.html>

I Test reports

mediAkit was tested by several test users before the final release.

Alpha-Test report from Thor

Jeg var forsøkskanin på Media Kit i går, og jeg har skrevet ned noen punkter som jeg mener må fikses.

1. Når du klikker på shellsriptet blir det lastet ned under navnet "modules.php", hvilket ikke stemmer overens med instruksjonene.³⁴
2. Det burde vært automatisk støtte i shellscripet for å logge inn som root, altså noe i stil med det xcdroast gjør.
3. Det burde vært en wizard istedenfor en masse vinduer som popper opp. I mange av disse vinduene er det heller ingen cancel-knapp.
4. Løsningen for apt-cdrom er noe Reodor Felgen-aktig. Det er ikke pent å starte en server uten at brukeren har bedt om det.
5. Scriptet gjør en update med en spesialversjon av "sources.list", men det tar ikke etterpå en ny update med den vanlige "sources.list". Dette medfører at apt-databasen blir feil.
6. Det bør legges inn et ikon på desktopen til den brukeren som startet scriptet.

Thor

Alpha-Test report from Henning

1. After installation the user should be offered information on what is installed, and how to get started, e.g. link to a howto page.
I HOWTO'en står det at det skal dukke opp et ikon på desktopen til den brukeren som installerer pakken. Siden pakken må installeres som root og det ikke er anbefalt at folk bruker maskinen som root hele tiden bør det forklares at ikonet også havner under "Extra" / "Lyd og Bilde" / "media Kit" ...
2. Når jeg har startet media-kit har jeg tre valg. Det nederste er opplagt, mplayer kan jeg kanskje gjette meg til er en mediaplayer, men kino ville jeg også gjette var en filmfremviser om jeg ikke hadde visst at det var noe annet. Red-Hat har skjønt poenget her: Folk er ikke interessert i å vite hva programmene heter, de vil vite hva de gjør. Derfor er det bedre å la valgene være noe slikt som: "Edit DV video", "Play video" og "Burn DVD".
3. Bug: Trykk mplayer. Mplayer starter. Lukk Mplayer. Får en beskjed om "Unable to start program".
4. Det jeg savner videre er litt info om hva jeg skal gjøre rent praktisk for å redigere en film. En forklaring av typen:
 - (a) Det første du gjør er å gå inn i kino, her gjør du sånn og sånn for å ta opp video, slik legger du ting i en sekvens og tilslutt må du eksportere til et bestemt format.
 - (b) Du har nå laget en film. Det kan være en ide å se på den før du brenner ut filmen. Dette kan gjøres med filmfremviseren. Åpne filen her og se på den. Var den ikke bra, gå tilbake til 1.
 - (c) Dersom filmen så bra ut kan du brenne den ut på DVD eller VCD. For å gjøre dette må du....

³⁴This problem was related to the postnuke installation. Since postnuke is replaced by gforge, this problem no longer relevant.

5. Følelsen jeg har er litt “hva nå”. Det er ikke helt klart hvilke formater man bør eksportere til, eller om det holder å lagre som ... Når programmene skal brukes sammen bør det forklares hvordan de henger sammen, hvilke formater som gir best kvalitet osv. ...

Å få alt installert hjelper en hel del. Det fjerner vanskeligheten med å installere programmer med masse avhengigheter osv. Men fordelene med å installere alt manuelt er at man selv må tenke ut hva man trenger. Når alt er installert må man få en slags forklaring på hva man trenger de forskjellige tingene til.

6. Kino:

```
BUG: modulen raw1394 lastes ikke automatisk....
```

```
BUG: raw1394 devicet er eid av root...
```

7. DVD-burner: Her virker det noe uklart hva man skal gjøre. Hva slags filer støtter den for eksempel? Jeg får feilmeldingen “Unable to burn”, uten noe mere spesifisering. Nå har jeg ikke noen plate i engang, men jeg vet jo ikke om jeg har eksportert riktig i kino engang ...

Her er output fra mediakit_dvd:

```
scsidev: '0,0,0'
```

```
scsibus: 0 target: 0 lun: 0
```

```
dvdrecord: Permission denied. Cannot open '/dev/sg0'. Cannot open SCSI driver.
```

```
dvdrecord: For possible targets try 'cdrecord -scanbus'. Make sure you  
are root.
```

```
ERROR BURN DVD: Unable to burn ISO-image Finished, sig =3D 17
```

```
crw-rw---- 1 root disk 21, 0 aug 31 2002 /dev/sg0
```

Som root får jeg kjørt cdrecord -scanbus, men det funker ikke som en vanlig bruker. Her må det også gjøres noe for å gi devicet til brukeren ved login ...

- Henning

Part IV

License

J Licenses for mediAkit

In the following the end user license agreement (EULA) and the License Boilerplate for **mediAkit** are shown.

J.1 The End User License Agreement (EULA)

LICENSE AGREEMENT AND LIMITED PRODUCT WARRANTY
Norsk Regnesentral "NR mediAkit Distribution"

Please read this document carefully before installing "NR mediAkit Distribution", any of its packages, or any software included with this product, on your computer. This document contains important information about your legal rights. By installing any or all of the software included with this product, you agree to the following terms and conditions.

GENERAL

As used herein, "EULA" means an end user license agreement, and "Software Programs" means, collectively, the Linux Programs and the Third-Party Programs, as each of those terms is defined herein. NR mediAkit Distribution consists of Linux packages made up of many individual software components, each of which was individually written and copyrighted. Throughout this document these components are referred to, individually and collectively, as the "OS Programs." Each OS Program has its own applicable end user license agreement. Most of the OS Programs are licensed pursuant to an open source EULA that permits you to copy, modify, and redistribute the software, in both source code and binary code forms. With the exception of the content of certain especially identified files, the remaining OS Programs are freeware or have been placed in the public domain. To understand the applicable EULA for each OS Program, your rights under it and to realize the maximum benefits available to you with the NR mediAkit Distribution, you must review the on-line documentation that accompanies each OS Program. Nothing in this license agreement limits your rights under, or grants you rights that supercede, the terms of any applicable EULA.

Programs of "NR mediAkit Distributions", including the setup procedure come with the "NR mediAkit License" (NRML), using a triple-license NRML/LGPL/GPL. The NRML permits Norsk Regnesentral to use contributions to mediAkit (i.e., software with this license) to use this software in other projects, possibly with other licensing (including projects with a non-open-source license). In particular, Norsk Regnesentral is permitted to use the setup routines and user interface which cannot be defined as stand-alone programs in connection with other projects, herein link, merge and distribution of parts of the software with other software. Contributions to mediAkit are only accepted with this triple license. The distributed software else is not affected by the NRML license.

The "NR mediAkit Distribution" may include an assortment of applications that might be licensed other than with an open source license. Throughout this document each of these software components are referred to, individually and collectively, as NOS-Programs." Generally, each of these NOS-Programs is licensed to you by the vendor pursuant to an end user license agreement ("NOS EULA") that generally permits you to install each of these products on only a single computer for your own individual use. Copying, redistribution, reverse engineering, and/or modification of these components may be prohibited, and you must look to the terms and conditions of the NOS EULA to determine your rights and any limitations imposed on you. Any violation by you of the applicable NOS EULA terms shall immediately terminate your license under that NOS EULA. For the precise terms of the NOS EULAs for each of these NOS Programs, please check the on-line documentation that accompanies each of them. If you do not agree to abide by the applicable license terms for these NOS Programs, then do not install them on your computer. If you wish to install any of these NOS Programs on more than one computer, please contact the vendor of the NOS Program to purchase additional licenses.

The NR mediAkit Distribution itself is a collective work under European / Norwegian copyright law. Subject to the trademark use limitations set forth in this Agreement, Norwegian Computing Center grants you a license in the collective work pursuant to the NRML/GPL/LGPL.

BEFORE INSTALLATION

CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE INSTALLING ANY OF THE SOFTWARE PROGRAMS. INSTALLING THE SOFTWARE PROGRAMS INDICATES YOUR ACCEPTANCE TO THE TERMS AND CONDITIONS SET FORTH IN THIS DOCUMENT AND OF THE END USER LICENSE AGREEMENT ASSOCIATED WITH THE SOFTWARE PROGRAM. IF YOU DO NOT AGREE WITH THESE TERMS AND CONDITIONS, DO NOT INSTALL THE SOFTWARE PROGRAMS.

THE SOFTWARE PROGRAMS, INCLUDING SOURCE CODE, DOCUMENTATION, APPEARANCE, STRUCTURE AND ORGANIZATION, ARE PROPRIETARY PRODUCTS OF NORSK REGNESENTRAL AND OTHERS AND ARE PROTECTED BY COPYRIGHT AND OTHER LAWS. TITLE TO THESE PROGRAMS, OR TO ANY COPY, MODIFICATION OR MERGED PORTION OF ANY OF THESE PROGRAMS, SHALL AT ALL TIMES REMAIN WITH THE AFOREMENTIONED, SUBJECT TO THE TERMS AND CONDITIONS OF THE APPLICABLE EULA RELATED TO THE SOFTWARE PROGRAMS UNDER CONSIDERATION.

NORSK REGNESENTRAL DOES NOT PROVIDE FREE SUPPORT FOR THE MEDIAKIT DISTRIBUTION OTHER THAN UPDATES THAT ARE PROVIDED WHENEVER IT MAY SUIT NORSK REGNESENTRAL. NORSK REGNESENTRAL IS NOT OBLIGATED TO PUBLISH NEW VERSIONS OF THE SOFTWARE OR CHANGES WITHIN A CERTAIN TIME FRAME.

LIMITED WARRANTY

EXCEPT AS SPECIFICALLY STATED IN THIS AGREEMENT OR IN AN EULA, THE SOFTWARE PROGRAMS ARE PROVIDED AND LICENSED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND

FITNESS FOR A PARTICULAR PURPOSE.

THE MEDIAKIT DISTRIBUTION IS PROVIDED AS A NETWORK SERVICE ONLY. NORSK REGNESENTRAL IS NOT RESPONSIBLE FOR PROBLEMS ACCURING FROM PROBLEMS CAUSED BY THE NETWORK (THE INTERNET) OR ONE OF ITS SERVICES.

Norsk Regnesentral does not warrant that the functions contained in the Software Programs will meet your requirements or that the operation of the Software Programs will be entirely error free or appear precisely as described in the accompanying documentation.

NORSK REGNESENTRAL DENIES ANY WARRANTY OR REMEDY OTHER THAN THOSE GIVEN BY NORWEGIAN LAWS.

LIMITATION OF REMEDIES AND LIABILITY

Norsk Regnesentral does not accept any liability whatsoever caused by the mediAkit distribution. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL NORSK REGNESENTRAL BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE PROGRAMS, EVEN IF NORSK REGNESENTRAL HAD BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

GENERAL

If any provision of this Agreement is held to be unenforceable, that shall not effect the enforceability of the remaining provisions. This Agreement shall be governed by the laws of Norway, without regard to any conflict of laws provisions. Copyright (C) 2003 Norwegian Computing Center. All rights reserved. "Linux" is a registered trademark of Linus Torvalds. All other trademarks are the property of their respective owners.

J.2 The License Boilerplate

```
/* ***** BEGIN LICENSE BLOCK *****
 * Version: NRML 1.0/GPL 2.0/LGPL 2.1
 *
 * The contents of this file are subject to the NR mediAkit License Version
 * 1.0 (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at http://mediakit.nr.no/NRML/
 *
 * Software distributed under the License is distributed on an "AS IS" basis,
 * WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License
 * for the specific language governing rights and limitations under the
 * License.
 *
 * The Original Code is _____
 *
 * The Initial Developer of the Original Code is
 * _____
 * Portions created by the Initial Developer are Copyright (C) 2___
 * the Initial Developer. All Rights Reserved.
 *
 * Contributor(s):
 *
 * Alternatively, the contents of this file may be used under the terms of
 * either the GNU General Public License Version 2 or later (the "GPL"), or
 * the GNU Lesser General Public License Version 2.1 or later (the "LGPL"),
 * in which case the provisions of the GPL or the LGPL are applicable instead
 * of those above. If you wish to allow use of your version of this file only
 * under the terms of either the GPL or the LGPL, and not to allow others to
 * use your version of this file under the terms of the NRML, indicate your
 * decision by deleting the provisions above and replace them with the notice
 * and other provisions required by the GPL or the LGPL. If you do not delete
 * the provisions above, a recipient may use your version of this file under
 * the terms of any one of the NRML, the GPL or the LGPL.
 *
 * ***** END LICENSE BLOCK ***** */
```

References

- [1] Bahner, Furu, Heen, Ingebrigtsen, Kvalnes, Reinholdtsen, Trondsen, Trosterud, Wisløff, Yrvin. Skolelinux-prosjektet. Web: <http://cvs.skolelinux.no/info/rapporter/forprosjekt.pdf>, 2002.
- [2] Ed Bailey. *Maximum RPM*. RedHat Press, 1997.
- [3] Brian Behlendorf. Open source as a business strategy. In Chris DiBona, Sam Ockman, and Mark Stone, editors, *Open Sources – Voices from the Open Source Revolution*. O'Reilly, 1999.
- [4] Robert J. Chassell and Richard M. Stallman. *Texinfo: The GNU Documentation Format, for Version 4*. GNU Press, 1999.
- [5] Robert de Leo. Self-hosting movies with movix. *Linux Journal*, www.linuxjournal.com, 7.april 2003, 2003.
- [6] Chris DiBona, Sam Ockman, and Mark Stone, editors. *Open Sources – Voices from the Open Source Revolution*. O'Reilly, 1999.
- [7] M. Dölle and B. Praschinger. Video-dvds selbst herstellen – selbst-gebrannt. *Linux Magazin*, 1:42–43, 2003.
- [8] D. Phillips F. Déchelle, G. Geiger. Demudi: The debian multimedia distribution. In *ICMC2001*. available on <http://www.demudi.org/doc/paper.pdf>, 2001.
- [9] S. Feldman. Make – a program for maintaining computer programs. *Software-Practice and Experience*, 9:225–265, 1979.
- [10] Joseph Feller and Brian Fitzgerald. A framework analysis of the open source software development paradigm. In *International Conference on Information Systems, Proceedings of the twenty first international conference on Information systems, 2000 , Brisbane, Queensland, Australia*, 2000.
- [11] Eleftherios Gkioulekas. *Developing software with GNU*. <http://www.amath.washington.edu/~lf/tutorials/autoconf/toolsmanual.dvi>, 1999. Note: several versions available on the Internet, e.g., <http://www.mars-attacks.org/~boklm/divers/doc/toolsmanual.pdf>.
- [12] GNU. Various Licenses and Comments about Them. *Web pages*, <http://www.gnu.org/philosophy/license-list.html>, 2002.
- [13] R. Scott Granneman. A very apropos apt. *Linux Magazine*, October 2003:24–28, 2003.
- [14] H. Hegna and W. Leister. Åpen kildekode (høringsuttalelse). In *Teknologirådet, høring om programvarepolitikk*. Teknologirådet, available on <http://www.teknologiradet.no/html/508.htm>, 2003. presentation available on http://www.teknologiradet.no/files/norsk_regnesentral.pdf.
- [15] W. Leister, S. Boudko, O. Aamot, and P. Holmes. Digital TV – A survey. Report 988, Norsk Regnesentral, Oslo, 2002.
- [16] W. Leister, E. Maus, B. Nordlund, O. Aamot, and J. Lous. Developing a Multimedia Distribution. Notat DART/02/02, Norsk Regnesentral, Oslo, 2002.
- [17] Petter Merok and Espen Andersen. Referat fra møte i pfit 2.oktober 2003: **Åpen kildekode – kilde til innovasjon eller døden for programvarebransjen**. <http://www.espen.com/pfit/PFIT20031002.pdf>, 2003.
- [18] Rich Morin, editor. *Prime Time Freeware for UNIX, Issue 2-2*. Prime Time Freeware, Sunnyvale, CA, July 1993.

- [19] J. Peek, T. O'Reilly, and M. Loukides, editors. *Unix Power Tools*. O'Reilly, Bantam Books, March 1993.
- [20] Bruce Perens. The open source definition. In Chris DiBona, Sam Ockman, and Mark Stone, editors, *Open Sources – Voices from the Open Source Revolution*. O'Reilly, 1999.
- [21] Bruce Perens and Debian. Debian Social Contract. *Web pages*, http://www.debian.org/social_contract, 1997.
- [22] Eric S. Raymond. *The Cathedral & the Bazaar*. O'Reilly, 1999.
- [23] Richard M. Stallman. *Free Software, Free Society: Selected Essays*. GNU Press, 2002.
- [24] Richard M. Stallman and Roland McGrath. *GNU Make: A Program for Directing Recompilation*. GNU Press, 2002.
- [25] Guy Steele and Eric S. Raymond. The Jargon File. *Web pages*, <http://www.tuxedo.org/~esr/jargon/>, 2000.
- [26] Free Standards Group Workgroup. Linux standard base specification 1.3. *Free Standards Group*, http://www.linuxbase.org/spec/refspecs/LSB_1.3.0/gLSB/gLSB.pdf, 2002.

Index

- ChangeLog, 62
- Makefile.am, 62
- autogen.sh, 62
- configure.ac, 62

- Advanced Authoring Format, 52
- AGNULA, 5, 57
- Alsaplayer, 50
- Anaconda, 34
- Ant, 24
- Apache, 35, 35
- APT, 28
- apt-get, 13
- APT-RPM, 29
- Ardour, 39
- Audacity, 42, 42
- autoconf, 22
- automake, 22

- BICK, 34
- Bug-buddy, 33
- Bugzilla, 33
- business model, 9

- cdparanoia, 53, 53
- cdrdao, 52
- cdrecord-prodvd, 53
- cdrtools, 53
- Cinelerra, 41
- CLM, 42
- CMN, 42
- Common Lisp Music, 42
- Common Music Notation, 42
- Coriander, 38, 38
- Current, 28

- DAP, 43, 43
- Debian, 11
- DeMuDi, 5, 57, 58
- detect, 31, 31
- digiKam, 38, 38
- DirectPresenter, 50
- discover, 31, 31
- distribution, 5, 7
- distro, 7
- Dpkg, 11, 25
- dvb-mpegtools, 45
- dvbstream, 54
- dvdauthor, 52
- dvdrttools, 53
- dvgrab, 37, 37

- EffecTV, 41, 41
- eMoviX, 5

- EPM, 26, 26
- ESP Package Manager, 26
- EXIF, 38

- Fedora, 19
- Festival, 56
- FFmpeg, 44
- Film Gimp, 41, 41
- Freevo, 51

- GAR, 24
- GForge, 35, 35
- GImageView, 51
- GNATS, 33
- Gnome-o-Phone, 55
- GnomeMeeting, 55
- GNU, 4
- GNUsound, 43, 43
- gPhoto, 40, 40
- gqview, 50, 50
- GStreamer, 37
- Gtkam, 38
- gtkam, 38
- gxine, 49

- harddrake, 31, 31
- Heroine Virtual: Cinelerra, 41
- HIØ DVB package, 54
- HTTPD, 35

- Icecast, 51
- ImageMagick, 48
- InstallBase, 29

- Jack audio connection kit, 40
- JFIF, 38

- Kernel Autoconfigure, 31, 31
- Kino, 41, 41
- kit, 7
- Knoppix, 5, 58
- kudzu, 31, 31

- lame, 45, 45
- liba52, 45
- libdv, 45
- libgphoto2, 38, 38
- libhardware, 32, 32
- libmpeg2, 44
- librtsp, 54
- libtool, 22
- LinDVD, 6
- Linphone, 55
- Linux, 4, 5, 54
- Linux Video Studio, 42, 42
- LinuxTV, 54

- LiveIce / LiveIce XMMS, 39
- Loki Patch Tools, 27
- Loki Setup Graphic Installer, 27
- Loki Update Tool, 27

- make, 21
- MEncoder, 44
- mjpegtools, 46
- Monkey Media, 48, 48
- MoviX, 58
- Mozilla, 4
- mpeg2dec, 47
- mpeg_play, 50
- mpg32ogg, 53
- MPlayer, 48, 58
- MSYS toolkit, 47
- MUSE, 43, 43
- MythTV, 51

- NIST DASE, 58
- NRH-up2date, 28

- Open Office, 4
- open source
 - development model, 4
 - distribution model, 4
- OpenOffice, 3

- PGI, 34
- PHP bugtracker, 33
- Planet CCRMA at Home, 58
- Plone, 35, 35
- PMS, 10
- Portage, 29
- Postnuke, 35, 35

- RAT, 55

- Red Hat, 19
- Red Hat Network Update Agent, 27
- ReHMuDi, 57
- Rezound, 43, 43
- Rhythmbox, 48, 48
- RPM, 12, 13, 25

- sampeg-2, 44
- SANE, 40
- savannah, 4
- Skolelinux, 4, 11
- SLab, 38
- smilutils, 48
- Software Packaging, 62
- Sorcery, 30
- sourceforge, 4
- SuSE, 19

- TerminatorX, 43, 43
- TIFF, 38
- transcode, 46

- up2date, 27

- vcdimager, 52
- vic, 56
- VideoLAN Client, 50
- VideoLAN Server, 51
- vsound, 40

- xawtv, 37, 37
- Ximian Red Carpet Client, 29
- Xine, 49
- XMMS, 49
- XMMS Recorder, 39
- XMovie, 49
- Xvoice, 56