# 1 Introduction

KRPMBuilder is a program that tries to aid users in building rpms. Allthough it may be a rather complicated process to build a rpm, only a few steps are to be considered for a lot of software packages. Version 1.2 uses four steps in a wizardlike fashion to set up the specfile, which controls the entire build process. If you want to use KRPMBuilder some basic knowledge on rpms and the way they are constructed, is needed. There will be some basic instructions on how to get things going in (3).

The specfile is built using a templatefile named standard.spec. It is not possible for now to control all the details of the build process. A version is planned for the future that will give a more sophisticated control to the user. If you need more options use krpmbuilder to construct the specfile and your favourite texteditor to modify the generated file as needed. Afterwards start rpm from the shell.

KRPMBuilder uses the standard sequence configure – make – make install to compile the package. If this is not possible with your source, KRPMBuilder will fail.

# 2 Using KRPMBuilder

Once you installed the program there should be an entry in your menu.

## 2.1 First startup

Starting KRPMBuilder for the first time you will be presented a dialogue to select the directories for the SPEC-files and the SOURCE-files. For more information on this see (3). This will change in one of the next versions, since it is impossible to use different paths for the directories needed.

## 2.2 First Page (Project Page)

This page of the wizard allows you to enter basic information about the package you want to create. Enter the project's name in the first editfield. This name together with the version will form the name of the sourcetarball to be processed <projectname>-<version>.tar.gz(bz2) should be the most common form.

The icons besides this edit field are used to create a new project, 

to open an existing project, 

to save a project, 

or to change the settings (configure the paths). 

The second box is used to enter the version of the project.

A rpm may be released for different distributions, using the same version of the sourcetarball. Each of this releases is assiged a unique number. This number has to be entered in the next inputfield.

Use the selectionbox to specify the license to be applied to the package.

Next specify the homepage of the package.

Enter your name in the edibox.

Finally selcet the path of the sourcetarball. When you start the build process the file will be copied from this location to the SOURCES-directory. After all information is

entered, you may select the next-Button to enter page two.

### 2.3 Second Page (Package Page)

More information on the package to be built is collected in this page.

Enter the distribution the package is built for in the first box. This corresponds to the releasenumber entered in the first page. If you supply a new package, that contains a patch, you will use the same combination auf release/distribution, since the version did not change.

Next you may enter the group this package will be assigned to. This is select-box that has to be supplied with more possible groups in the future.

The 'Provides' entry should give users of your package a hint on the purpose of the software.

Enter the names of required libs etc. in the next box.

Vendor takes the name of the software's developer.

The next two boxes should be used to give a shorthand description and a more precise one of what the software can be used for.

### 2.4 Third Page (Build Page)

This page collects informations needed for the build process itself. If you want to change the build-root, you may enter this information in the first editfield. The build-root specifies where temporary information is stored during the creation of the package.

Use the next two boxes to specify the prefix for the build operation and any options to be supplied for the configure step.

The selection box is used to name the architecture the package is constructed for.

Finally you may decide to build only a binary package or both a binary and a source package.

### 2.5 Fourth Page (Finish Page)

The last page controls the buildprocess itself. Pressing the start-button makes the progam check for all neccessary information, set up the specfile and start rpm. You may stop the process any time using the stop-button. After successful operation, the packages are created in the RPMS and SRPM-directories.

## 3 Installation

This Chapter will help you in the process of obtaining and installing KRPMBuilder.

### 3.1 How to obtain KRPmbuilder

KRPMBuilder is hosted on sourceforge.net. To get the files needed for Installation, direct your browser to the following location:
http://sourceforge.net/projects/krpmbuilder/

Here you will find a link to the donwloads of KRPMbuilder which consist of a sourcetarball, a rpm for SuSE 9.1 and a sourcerpm.

### 3.2 Compiling

Extract the files of the tar-archive into a directory of your choice. Use the ./configure

command to build the Makefiles for KRPMBuilder. After that a make and make install should be sufficent to get a working KRPMBuilder. The development environment of kde version 3.2 is needed for a successfull compilation. You will need root permissions to do the make install.

If you want to use KRPMBuilder to set things up, change to the ./src directory and call ./krpmbuilder. This will allow you to create a rpm-package for KRPMBuilder, which in turn can be used to install everything.

Please report any problems to the appropriate forums of the project.

# 4 About rpms

This chapter will show you some aspects of rpms. For a more complete documentation visit: http://www.rpm.org/ If you want to define rpm-macros they must reside in the file '.rpmmacros' in your home-directory.

## 4.1 Directory structure

In order to successfully build and use rpm's you need the following directories:

- SPECS this is where the specfile will be located
- SOURCES the source tarball will be copied here
- BUILD the build takes place here
- SRPMS the sourcerpm will be build in this folder
- RPMS the binary package will be build here

Depending on the type of distribution this directory structure is located in '/user/src/packages' for SuSE Linux and in '/usr/src/redhat' for Fedora and Red Hat Linux. These folders allow write access to the root user only. So you might want to use directories in  /home/mypath/rpm. You need to specify the macro %_topdir '/home/mypath/rpm' in ~/.rpmmacros then. If you forget to do so, the building process will be unable to find the source for the build.