NAME

glilypond — integrate lilypond parts into groff

SYNOPSIS

glilypond [*options*] [**-**] [**--**] [*filespec* ...]

DESCRIPTION

glilypond transforms sheet music written in the *lilypond* language into the groff(7) language using the **.PSPIC** request, such that groff(1) can transform it into a format that can be displayed directly.

Files in *groff* language and *standard input* can be provided as arguments.

OPTION OVERVIEW

Breaking Options

-?|-h|--help|--usage
: Print help or usage information, then leave the program.

--version
: Print version information.

-l|--license
: Print license information.

Options for building EPS Files

[**--ly2eps**]
: Here the **lilypond** program creates *eps* files directly. This is the default.

[**--pdf2eps**]
: The program **glilypond** generates a *pdf* file using **lilypond**. Then the*eps* file is generated by **pdf2ps** and **ps2eps**R**.**

Directories and Files

-e|--eps_dir *directory_name*
: Normally all *EPS* files are sent to the temporary directory. With this option, you can generate your own directory, in which all useful *EPS* files are send. So at last, the temporary directory can be removed.

-p|--prefix *begin_of_name*
: Normally all temporary files get names that start with the **ly**... prefix. With this option, you can freely change this prefix.

-k|--keep_all
: Normally all temporary files without the *eps* files are deleted. With this option, all generated files either by the **lilypond** program or other format transposers are kept.

-t|--temp_dir *dir*
: With this option, you call a directory that is the base for the temporary directory. This directory name is used as is without any extensions. If this directory does not exist it is be created. The temporary directory is created by Perl's security operations directly under this directory. In this temporary directory, the temporary files are stored.

Output

-o|--output *file_name*
: Normally all *groff* output of this program is sent to **STDOUT**R**.** With this option, that can be changed, such that the output is stored into a file named in the option argument *file_name*.

-v|-V|--verbose
: A lot more of information is sent to STDERR.

Short Option Collections

The argument handling of options

Short options are arguments that start with a single dash **-**. Such an argument can consist of arbitrary many options without option argument, composed as a collection of option characters following the single dash.

Such a collection can be terminated by an option character that expects an option argument. If this option character is not the last character of the argument, the following final part of the argument is the option argument. If it is the last character of the argument, the next argument is taken as the option argument.

This is the standard for *POSIX* and *GNU* option management.

For example,

-kVe *some_dir*
: is a collection of the short options **-k** and **-V** without option argument, followed by the short option **-e** with option argument that is the following part of the argument *some_dir*. So this argument could also be written as several arguments **-k -V -e** *some_dir*.

Handling of Long Options

Arguments that start with a double dash **--** are so-called *long options R .* Each double dash argument can only have a single long option.

Long options have or have not an option argument. An option argument can be the next argument or can be appended with an equal sign **=** to the same argument as the long option.

--help is a long option without an option argument.

--eps_dir *some_dir*
--eps_dir=*some_dir*
: is the long option **--eps_dir** with the option argument *some_dir*.

Moreover the program allows abbreviations of long options, as much as possible.

The *long option* **--keep_all** can be abbreviated from **--keep_al** up to **--k** because the program does not have another *long option* whose name starts with the character **k**.

On the other hand, the option **--version** cannot be abbreviated further than **--vers** because there is also the *long option* **--verbose** that can be abbreviated up to **--verb**.

An option argument can also be appended to an abbreviation. So is **--e**=*some_dir* the same as **--eps_dir** *some_dir*.

Moreover the program allows an arbitrary usage of upper and lower case in the option name. This is *Perl* style.

For example, the *long option* **--keep_all** can as well be written as **--Keep_All** or even as an abbreviation like **--KeE**.

FILESPEC ARGUMENTS

An argument that is not an option or an option argument is called a *filespec* argument.

Without any *filespec* argument, *standard input* is read.

Each *filespec* argument must either be the name of a readable file or a dash **-** for *standard input*. Each input must be written in the *roff* or *groff* language and can include *lilypond* parts.

Normally arguments starting with a dash **-** are interpreted as an option. But if you use an argument that consists only of a doubled dash **-- R ,** all following arguments are taken as *filespec* argument, even if such an argument starts with a dash. This is handled according to the *GNU* standard.

THE LILYPOND PARTS IN ROFF INPUT

Integrated Lilypond Codes

A *lilypond* part within a structure written in the *groff* language is the whole part between the marks

.lilypond start

and

.lilypond end

A *groff* input can have several of these *lilypond* parts.

When processing such a *lilypond* part between **.lilypond start** and **.lilypond end** we say that the **glilypond** program is in *lilypond mode*.

These *lilypond* parts are sent into temporary *lilypond* files with the file name extension **.ly**. These files are transformed later on into *EPS* files.

Inclusion of ly-Files

An additional command line for file inclusion of *lilypond* files is given by
.lilypond include*file_name*
in *groff* input. For each such *include* command, one file of *lilypond* code can be included into the *groff* code. Arbitrarily many of these commands can be included in the *groff* input.

These include commands can only be used outside the *lilypond* parts. Within the *lilypond mode*, this inclusion is not possible. So **.lilypond include** may not be used in *lilypond mode*, i.e. between **.lilypond start** and **.lilypond end**. These included*ly*-files are also transformed in to *EPS* files.

GENERATED FILES

By the transformation process of *lilypond* parts into *EPS* files, there are many files generated. By default, these files are regarded as temporary files and as such stored in a temporary directory.

This process can be changed by command line options.

Command Line Options for Directories

The temporary directory for this program is either created automatically or can be named by the option **-t|--temp_dir** *dir*.

Moreover, the *EPS* files that are later on referred by **.PSPIC** command in the final *groff* output can be stored in a different directory that can be set by the command line option **-e|--eps_dir** *directory_name*. With this option, the temporary directory can be removed completely at the end of the program.

The beginning of the names of the temporary files can be set by the command line option [**-p** |] *begin_of_name*.

All of the temporary files except the *EPS* files are deleted finally. This can be changed by setting the command line option [**-k** |] With this, all temporary files and directories are kept, not deleted.

These *EPS* files are stored in a temporary or *EPS* directory. But they cannot be deleted by the transformation process because they are needed for the display which can take a long time.

TRANSFORMATION PROCESSES FOR GENERATING EPS FILES

Mode ly2eps

This mode is the default. It can also be chosen by the option **--ly2eps**.

In this mode, the **.ly** files are transformed by the **lilypond** program into many files of different formats, including *eps* files, using

$ lilypond --ps -dbackend=eps -dgs-load-fonts --output=*file-name*

for each **.ly** file. The output *file-name* must be provided without an extension, its directory is temporary.

There are many *EPS* files created. One having the complete transformed **ly** file, named *file-name***.eps**.

Moreover there are *EPS* files for each page, named *file-name*-*digit*.**eps**.

The last step to be done is replacing all *lilypond* parts by the collection of the corresponding *EPS* page files. This is done by *groff* commands
.PSPIC *file-name*-*digit*.**eps**

Mode pdf2eps

This mode can be chosen by the option **--pdf2eps**.

In this mode, the **.ly** files are transformed by the **lilypond(1)** program into *pdf* files, using
lilypond --pdf --output=*file-name*
for each **.ly** file. The *file-name* must be provided without the extension **.pdf**. By this process, a file *file-name*.**pdf** is generated.

The next step is to transform these *PDF* files into a *PS* file. This is done by the **pdf2ps(1)** program using
$ **pdf2ps** *file-name* **.pdf** *file-name* **.ps**
The next step creates an *EPS* file from the *PS* file. This is done by the **ps2eps(1)** program using
$ **ps2eps** *file-name*.**ps**

By that, a file *file-name*.**eps** is created for each *lilypond* part in the *groff* file or standard input.

The last step to be done is replacing all *lilypond* parts by the *groff* command
.PSPIC *file-name*.**eps**

THE GENERATED NEW ROFF STRUCTURE

The new groff(7) structure generated by **glilypond** is either

1) sent to standard output and can there be saved into a file or piped into groff(1) or grofer(1) or

2) stored into a file by given the option **-o** | --output *file_name*

SEE ALSO

groff(1) the usage of the groff program and pointers to the documentation and availability of the *groff* system. The main source of information for the *groff* system is the *groff* **info(1)** file.

groff(7) documents the *groff* language.

groff_tmac(5) contains documentation of the **.PSPIC** request.

lilypond(1)
The documentation of the **lilypond** program. The main source of information for the *lilypond* language is the *lilypond* **info(1)** file.

pdf2ps(1)
transform a *PDF* file into a *Postscript* format.

ps2eps(1)
transform a *PS* file into an *EPS* format.

COPYING

AUTHORS

This document was written by Bernd Warken.