

LilyPond

The music typesetter

The LilyPond development team

Copyright © 1999–2005 by the authors

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled “GNU Free Documentation License”.

(For LilyPond version 2.6.5)

Table of Contents

Preface	1
Notes for version 2.6.....	1
1 Introduction	2
1.1 Engraving	2
1.2 Automated engraving.....	3
1.3 What symbols to engrave?	5
1.4 Music representation.....	6
1.5 Example applications.....	7
1.6 About this manual	8
2 Tutorial	11
2.1 First steps	11
2.2 Running LilyPond for the first time.....	12
MacOS X.....	12
Windows	13
Unix	13
2.3 More about pitches.....	13
2.4 Entering ties.....	14
2.5 Automatic and manual beams.....	15
2.6 Octave entry.....	16
2.7 Music expressions explained.....	17
2.8 More staves.....	19
2.9 Adding articulation marks to notes.....	20
2.10 Combining notes into chords.....	22
2.11 Advanced rhythmic commands.....	22
2.12 Commenting input files.....	23
2.13 Printing lyrics.....	23
2.14 A lead sheet.....	24
2.15 Adding titles.....	25
2.16 Single staff polyphony.....	26
2.17 Piano staves.....	27
2.18 Organizing larger pieces.....	27
2.19 An orchestral part.....	28
3 Example templates	30
3.1 Single staff.....	30
3.1.1 Notes only.....	30
3.1.2 Notes and lyrics.....	30
3.1.3 Notes and chords.....	31
3.1.4 Notes, lyrics, and chords.....	32
3.2 Piano templates.....	32
3.2.1 Solo piano.....	32
3.2.2 Piano and melody with lyrics.....	33
3.2.3 Piano centered lyrics.....	34
3.2.4 Piano centered dynamics.....	35
3.3 String quartet.....	37

3.3.1	String quartet	37
3.3.2	String quartet parts	39
3.4	Vocal ensembles	41
3.4.1	SATB vocal score	41
3.4.2	SATB vocal score and automatic piano reduction	42
3.5	Ancient notation templates	44
3.5.1	Transcription of mensural music	45
3.6	Jazz combo	50
3.7	Other templates	55
3.7.1	All headers	56
3.7.2	Gregorian template	57
3.8	Lilypond-book templates	57
3.8.1	LaTeX	57
3.8.2	Texinfo	58
4	Putting it all together	59
4.1	Suggestions for writing LilyPond files	59
4.2	Extending the templates	59
4.3	Fixing overlapping notation	62
5	Running LilyPond	64
5.1	Invoking lilypond	64
5.2	Command line options	64
5.3	Environment variables	67
5.4	Error messages	67
5.5	Updating with <code>convert-ly</code>	68
5.6	Reporting bugs	69
5.7	Editor support	70
5.8	File structure	70
5.9	Including LilyPond files	71
6	Basic notation	73
6.1	Note entry	73
6.1.1	Notes	73
6.1.2	Pitches	73
6.1.3	Cautionary accidentals	74
6.1.4	Micro tones	75
6.1.5	Chords	75
6.1.6	Rests	75
6.1.7	Skips	76
6.1.8	Durations	76
6.1.9	Augmentation dots	77
6.1.10	Tuplets	77
6.1.11	Scaling durations	78
6.2	Alternate music entry	78
6.2.1	Relative octaves	79
6.2.2	Octave check	80
6.2.3	Transpose	80
6.2.4	Bar check	81
6.2.5	Skipping corrected music	82
6.2.6	Automatic note splitting	82
6.3	Staff notation	83
6.3.1	Clef	83

6.3.2	Key signature	84
6.3.3	Time signature	85
6.3.4	Partial measures	86
6.3.5	Bar lines	87
6.3.6	Unmetered music	88
6.3.7	System start delimiters	89
6.3.8	Staff symbol	90
6.4	Connecting notes	91
6.4.1	Ties	91
6.4.2	Slurs	92
6.4.3	Phrasing slurs	93
6.4.4	Automatic beams	93
6.4.5	Manual beams	94
6.4.6	Grace notes	95
6.5	Expressive marks	97
6.5.1	Articulations	97
6.5.2	Fingering instructions	99
6.5.3	Dynamics	101
6.5.4	Breath marks	102
6.5.5	Running trills	102
6.5.6	Glissando	103
6.5.7	Arpeggio	103
6.6	Polyphony	104
6.6.1	Basic polyphony	105
6.6.2	Explicitly instantiating voices	106
6.6.3	Collision Resolution	108
6.7	Repeats	109
6.7.1	Repeat types	109
6.7.2	Repeat syntax	109
6.7.3	Repeats and MIDI	111
6.7.4	Manual repeat commands	112
6.7.5	Tremolo repeats	112
6.7.6	Tremolo subdivisions	113
6.7.7	Measure repeats	113
7	Instrument-specific notation	115
7.1	Piano music	115
7.1.1	Automatic staff changes	115
7.1.2	Manual staff switches	116
7.1.3	Pedals	116
7.1.4	Staff switch lines	117
7.1.5	Cross staff stems	118
7.2	Chord names	118
7.2.1	Introducing chord names	118
7.2.2	Chords mode	119
7.2.3	Printing chord names	121
7.3	Vocal music	124
7.3.1	Setting simple songs	124
7.3.2	Entering lyrics	125
7.3.3	Hyphens and extenders	126
7.3.4	The Lyrics context	127
7.3.5	Flexibility in alignment	129
7.3.5.1	Lyrics to multiple notes of a melisma	129
7.3.6	Switching the melody associated with a lyrics line	130

7.3.7	More stanzas	131
7.3.8	Ambitus	133
7.3.9	Other vocal issues	134
7.4	Rhythmic music	134
7.4.1	Showing melody rhythms	134
7.4.2	Entering percussion	134
7.4.3	Percussion staves	135
7.5	Guitar	137
7.5.1	String number indications	137
7.5.2	Tablatures basic	137
7.5.3	Non-guitar tablatures	138
7.5.4	Fret diagrams	139
7.5.5	Other guitar issues	140
7.6	Bagpipe	140
7.6.1	Bagpipe definitions	140
7.6.2	Bagpipe example	141
7.7	Ancient notation	142
7.7.1	Ancient note heads	142
7.7.2	Ancient accidentals	143
7.7.3	Ancient rests	143
7.7.4	Ancient clefs	144
7.7.5	Ancient flags	146
7.7.6	Ancient time signatures	147
7.7.7	Ancient articulations	148
7.7.8	Custodes	148
7.7.9	Divisiones	149
7.7.10	Ligatures	150
7.7.10.1	White mensural ligatures	150
7.7.10.2	Gregorian square neumes ligatures	151
7.7.11	Gregorian Chant contexts	156
7.7.12	Mensural contexts	157
7.7.13	Figured bass	157
7.8	Other instrument specific notation	159
7.8.1	Artificial harmonics (strings)	159
8	Advanced notation	160
8.1	Text	160
8.1.1	Text scripts	160
8.1.2	Text spanners	160
8.1.3	Text marks	161
8.1.4	Text markup	162
8.1.5	Text encoding	164
8.1.6	Nested scores	165
8.1.7	Overview of text markup commands	165
8.1.8	Font selection	172
8.1.9	New dynamic marks	173
8.2	Preparing parts	173
8.2.1	Multi measure rests	173
8.2.2	Metronome marks	175
8.2.3	Rehearsal marks	176
8.2.4	Bar numbers	177
8.2.5	Instrument names	178
8.2.6	Instrument transpositions	179
8.2.7	Ottava brackets	180

8.2.8	Different editions from one source	180
8.3	Orchestral music	182
8.3.1	Automatic part combining	182
8.3.2	Hiding staves	183
8.3.3	Quoting other voices	184
8.3.4	Formatting cue notes	185
8.3.5	Aligning to cadenzas	186
8.4	Contemporary notation	186
8.4.1	Polymetric notation	187
8.4.2	Time administration	189
8.4.3	Clusters	189
8.4.4	Special fermatas	190
8.4.5	Special noteheads	190
8.4.6	Feathered beams	191
8.4.7	Improvisation	191
8.5	Educational use	192
8.5.1	Balloon help	192
8.5.2	Blank music sheet	192
8.5.3	Hidden notes	193
8.5.4	Shape note heads	193
8.5.5	Easy Notation note heads	194
8.5.6	Analysis brackets	194
8.5.7	Coloring objects	195
8.6	Automatic notation	196
8.6.1	Automatic accidentals	196
8.6.2	Setting automatic beam behavior	198
9	Changing defaults	201
9.1	Interpretation contexts	201
9.1.1	Creating contexts	202
9.1.2	Changing context properties on the fly	203
9.1.3	Modifying context plug-ins	205
9.1.4	Layout tunings within contexts	206
9.1.5	Changing context default settings	207
9.1.6	Defining new contexts	208
9.2	The <code>\override</code> command	210
9.2.1	Common tweaks	210
9.2.2	Constructing a tweak	212
9.2.3	Navigating the program reference	212
9.2.4	Layout interfaces	213
9.2.5	Determining the grob property	214
9.2.6	Difficult tweaks	215
10	Output formats	217
10.1	Paper output	217
10.1.1	Setting global staff size	217
10.1.2	Selecting notation font size	217
10.1.3	Paper size	218
10.1.4	Page formatting	219
10.1.5	Score layout	221
10.1.6	Vertical spacing	221
10.1.7	Vertical spacing of piano staves	222
10.1.8	Horizontal Spacing	223

10.1.9	Line length	224
10.1.10	Line breaking	224
10.1.11	Page breaking	225
10.1.12	Multiple movements	225
10.1.13	Creating titles	226
10.2	Sound output	229
10.2.1	Creating MIDI files	229
10.2.2	MIDI block	230
10.2.3	MIDI instrument names	230
11	Interfaces for programmers	231
11.1	Programmer interfaces for input	231
11.1.1	Input variables and Scheme	231
11.1.2	Internal music representation	231
11.1.3	Extending music syntax	232
11.1.4	Manipulating music expressions	233
11.1.5	Displaying music expressions	234
11.1.6	Using LilyPond syntax inside Scheme	234
11.2	Markup programmer interface	236
11.2.1	Markup construction in Scheme	236
11.2.2	How markups work internally	237
11.2.3	Markup command definition	237
11.3	Contexts for programmers	239
11.3.1	Context evaluation	239
11.3.2	Running a function on all layout objects	239
12	lilypond-book: Integrating text and music	240
12.1	An example of a musicological document	240
12.2	Integrating LaTeX and music	243
12.3	Integrating Texinfo and music	244
12.4	Integrating HTML and music	245
12.5	Music fragment options	245
12.6	Invoking lilypond-book	247
12.7	Filename extensions	248
13	Converting from other formats	249
13.1	Invoking midi2ly	249
13.2	Invoking etf2ly	250
13.3	Invoking abc2ly	250
13.4	Invoking mup2ly	251
13.5	Generating LilyPond files	251
Appendix A	Literature list	252
Appendix B	Scheme tutorial	253
Appendix C	Notation manual details	255
C.1	Chord name chart	255
C.2	MIDI instruments	256
C.3	List of colors	257
C.4	The Feta font	259

Appendix D	Point and click	271
Appendix E	Cheat sheet	272
Appendix F	GNU Free Documentation License	276
	F.0.1 ADDENDUM: How to use this License for your documents	281
Appendix G	LilyPond index	282

Preface

It must have been during a rehearsal of the EJE (Eindhoven Youth Orchestra), somewhere in 1995 that Jan, one of the cranked violists told Han-Wen, one of the distorted French horn players, about the grand new project he was working on. It was an automated system for printing music (to be precise, it was MPP, a preprocessor for MusiXTeX). As it happened, Han-Wen accidentally wanted to print out some parts from a score, so he started looking at the software, and he quickly got hooked. It was decided that MPP was a dead end. After lots of philosophizing and heated email exchanges, Han-Wen started LilyPond in 1996. This time, Jan got sucked into Han-Wen's new project.

In some ways, developing a computer program is like learning to play an instrument. In the beginning, discovering how it works is fun, and the things you cannot do are challenging. After the initial excitement, you have to practice and practice. Scales and studies can be dull, and if you are not motivated by others—teachers, conductors or audience—it is very tempting to give up. You continue, and gradually playing becomes a part of your life. Some days it comes naturally, and it is wonderful, and on some days it just does not work, but you keep playing, day after day.

Like making music, working on LilyPond can be dull work, and on some days it feels like plodding through a morass of bugs. Nevertheless, it has become a part of our life, and we keep doing it. Probably the most important motivation is that our program actually does something useful for people. When we browse around the net we find many people who use LilyPond, and produce impressive pieces of sheet music. Seeing that feels unreal, but in a very pleasant way.

Our users not only give us good vibes by using our program, many of them also help us by giving suggestions and sending bug reports, so we would like to thank all users that sent us bug reports, gave suggestions or contributed in any other way to LilyPond.

Playing and printing music is more than a nice analogy. Programming together is a lot of fun, and helping people is deeply satisfying, but ultimately, working on LilyPond is a way to express our deep love for music. May it help you create lots of beautiful music!

Han-Wen and Jan

Utrecht/Eindhoven, The Netherlands, July 2002.

Notes for version 2.6

For years, LilyPond has been associated with T_EX, for its design, syntax and, last but not least, since it used T_EX as an output engine. Starting with 2.6, the latter has changed. By default, LilyPond now produces PostScript directly. This makes it easier to install, quicker to operate and more versatile.

Under the hood, this was made possible by use of the Pango library, which does typesetting of multilingual text. This means that you can easily typeset Chinese, Russian or Minoic lyrics. Another result is the SVG output. You can create SVG pictures of music notation directly from LilyPond.

There are also small improvements. This release has numerous extra features, such as color support, string-number notation, arrowed glissandi. Moreover, it is now possible to commission features. For a small fee, we (the core developers) can implement the features that you sorely need. Examples of sponsored features in 2.6 are solfa notation, stemlets, starting and stopping staves.

Han-Wen and Jan

Utrecht/Eindhoven, The Netherlands, May 2005.