

# LilyPond

---

Das Notensatzprogramm

## Benutzerhandbuch

### Das LilyPond-Entwicklerteam

Copyright © 1999–2007 bei den Autoren

*The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.*

*Die Übersetzung der folgenden Lizenzanmerkung ist zur Orientierung für Leser, die nicht Englisch sprechen. Im rechtlichen Sinne ist aber nur die englische Version gültig.*

Es ist erlaubt, dieses Dokument unter den Bedingungen der GNU Free Documentation Lizenz (Version 1.1 oder spätere, von der Free Software Foundation publizierte Versionen, ohne invariante Abschnitte), zu kopieren, verbreiten und/oder zu verändern. Eine Kopie der Lizenz ist im Abschnitt “GNU Free Documentation License” angefügt.

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”.

# Inhaltsverzeichnis

<b>1</b>	<b>Musikalische Notation</b>	<b>1</b>
1.1	Tonhöhen	1
1.1.1	Tonhöhen setzen	1
	Absolute Oktavenbezeichnung	1
	Relative Oktavenbezeichnung	2
	Versetzungsszeichen	3
	Notenbezeichnungen in anderen Sprachen	5
1.1.2	Viele Tonhöhen gleichzeitig verändern	6
	Oktavenüberprüfung	6
	Transposition	7
1.1.3	Tonhöhen anzeigen lassen	8
	Notenschlüssel	8
	Tonartbezeichnung	9
	Oktavierungsklammern	10
	Transposition von Instrumenten	10
	Automatische Versetzungsszeichen	10
	Tonumfang	10
1.1.4	Notenköpfe	10
	Besondere Notenköpfe	10
	Easy-Notation-Notenköpfe	10
	Notenköpfe mit besonderen Formen	10
	Improvisation	10
1.2	Rhythmus	11
1.2.1	Rhythmen eingeben	11
1.2.1.1	Tondauern	11
1.2.1.2	Andere rhythmische Aufteilungen	12
1.2.1.3	Tondauern skalieren	14
1.2.1.4	Bindebögen	15
1.2.2	Pausen eingeben	16
1.2.2.1	Pausen	17
1.2.2.2	Unsichtbare Pausen	17
1.2.2.3	Ganztaktige Pausen	18
1.2.3	Rhythmen anzeigen lassen	18
1.2.3.1	Taktangabe	18
1.2.3.2	Auftakte	19
1.2.3.3	Musik ohne Metrum	20
1.2.3.4	Polymetrische Notation	20
1.2.3.5	Automatische Aufteilung von Noten	21
1.2.3.6	Melodierhythmus anzeigen	21
1.2.4	Balken	22
1.2.4.1	Automatische Balken	22
1.2.4.2	Einstellung von automatischen Balken	22
1.2.4.3	Manuelle Balken	22
1.2.4.4	Gespreizte Balken	23
1.2.5	Takte	23
1.2.5.1	Taktlinien	24
1.2.5.2	Taktzahlen	25
1.2.5.3	Takt- und Taktzahlüberprüfung	25

1.2.5.4	Übungszeichen	26
1.2.6	Besondere rhythmische Fragen	26
1.2.6.1	Verzierungen	26
1.2.6.2	An Kadenzen ausrichten	29
1.2.6.3	Verwaltung der Zeiteinheiten	29
1.3	Ausdrucksbezeichnungen	29
1.3.1	An Noten angehängt	29
	Artikulationszeichen und Verzierungen	29
	Dynamik	32
	Neue Lautstärkezeichen	34
1.3.2	Bögen	34
	Legatobögen	34
	Phrasierungsbögen	35
	Laissez-vibrer-Bögen	35
	Atemzeichen	36
	Glissando zu unbestimmter Tonhöhe	36
1.3.3	Linien	36
	Glissando	36
	Arpeggio	37
	Triller	38
1.4	Wiederholungszeichen	39
1.4.1	Lange Wiederholungen eingeben	39
	Wiederholungstypen	39
	Die Syntax der Wiederholungen	40
	Manuelle Wiederholungsbefehle	42
1.4.2	Kurze Wiederholungen	43
	Prozent-Wiederholungen	43
	Tremolo-Wiederholung	44
	Tremolo-Unterteilung	44
1.5	Gleichzeitig erscheinende Noten	45
1.5.1	Eine einzelne Stimme	45
	Noten mit Akkorden	45
	Cluster	45
1.5.2	Mehrere Stimmen	45
	Einfache Mehrstimmigkeit	45
	Stimmen explizit beginnen	47
	Auflösung von Zusammenstößen	49
	Automatische Kombination von Stimmen	51
	Musik parallel notieren	51
1.6	Notation innerhalb eines Systems	53
1.6.1	Systeme anzeigen lassen	53
	Klammern am Systemanfang	53
	Das Notensystem	55
	Systeme verstecken	56
1.6.2	Orchesterstimmen erstellen	56
	Metronomangabe	56
	Instrumentenbezeichnungen	56
	Stichnoten	56
	Stichnoten formatieren	56
1.7	Anmerkungen	56
1.7.1	Innerhalb des Systems	56
	Auswahl der Notations-Schriftgröße	56
	Fingersatzanweisungen	56
	Unsichtbare Noten	57

Farbige Objekte .....	57
Klammern .....	57
Häse .....	57
1.7.2 Außerhalb des Notensystems .....	58
Erklärungen in Ballonform .....	58
Gitternetzlinien .....	58
Analyseklammern .....	58
1.8 Text .....	58
1.8.1 Text eingeben .....	58
1.8.1.1 Textarten .....	58
1.8.1.2 Text mit Verbindungslinien .....	58
1.8.1.3 Textartige Zeichen .....	58
1.8.1.4 Separater Text .....	58
1.8.2 Text formatieren .....	58
1.8.2.1 Textbeschriftung (Einleitung) .....	58
1.8.2.2 Überblick über die wichtigsten Textbeschriftungsbefehle .....	58
1.8.2.3 Textausrichtung .....	58
1.8.2.4 Graphische Notation innerhalb einer Textbeschriftung .....	58
1.8.2.5 Musikalische Notation innerhalb einer Textbeschriftung .....	58
1.8.2.6 Textbeschriftung über mehrere Seiten .....	58
1.8.3 Schriftarten .....	59
1.8.3.1 Schriftart des gesamten Dokuments .....	59
1.8.3.2 Schriftarten für einen Eintrag .....	59

## 2 Spezielle Notation ..... 60

2.1 Notation von Gesang .....	60
2.1.1 Übliche Notation für Gesang .....	60
2.1.1.1 Referenz für Musiktexte .....	60
2.1.1.2 Einfache Lieder setzen .....	60
2.1.1.3 Eingabe von Text .....	60
2.1.1.4 Mit Gesangstexten und Bezeichnungen arbeiten .....	60
2.1.2 Text an einer Melodie ausrichten .....	60
2.1.2.1 Automatische Silbendauer .....	60
2.1.2.2 Manuelle Silbendauer .....	60
2.1.2.3 Mehrere Silben zu einer Note .....	60
2.1.2.4 Mehrere Noten zu einer Silbe .....	60
2.1.2.5 Noten überspringen .....	60
2.1.2.6 Unterstrichen und Trennstriche .....	60
2.1.2.7 Gesangstext und Wiederholungen .....	60
2.1.3 Positionierung von Gesangstext .....	60
2.1.3.1 Getrennte Texte .....	60
2.1.3.2 Text unabhängig von den Noten .....	60
2.1.3.3 Hymnen .....	60
2.1.3.4 Silben platzieren .....	60
2.1.3.5 Gesangstext zwischen Systemen zentrieren .....	60
2.1.4 Strophen .....	60
2.1.4.1 Strophennummern hinzufügen .....	60
2.1.4.2 Lautstärkebezeichnung zu Strophen hinzufügen .....	60
2.1.4.3 Sängernamen zu Strophen hinzufügen .....	60
2.1.4.4 Strophen mit unterschiedlichem Rhythmus .....	60
2.1.4.5 Die Strophen am Ende ausdrucken .....	61
2.1.4.6 Die Strophen am Ende in mehreren Spalten drucken .....	61
2.2 Tasteninstrumente .....	61
2.2.1 Übliche Notation für Tasteninstrumente .....	61

Referenz für Tasteninstrumente .....	61
Notensysteme manuell verändern .....	61
Automatischer Systemwechsel .....	61
Stimmführungslinien .....	62
Häse über beide Systeme .....	63
2.2.2 Klavier .....	63
Klavierpedal .....	63
2.2.3 Akkordion .....	65
Diskant-Symbole .....	65
2.3 Bundlose Saiteninstrumente .....	65
2.3.1 Übliche Notation für nichtgebundene Saiteninstrumente .....	65
2.3.1.1 Hinweise für ungebundene Saiteninstrumente .....	65
2.3.2 Streichinstrumente .....	65
2.3.2.1 Referenz für Streicher .....	65
2.3.3 Zupfinstrumente .....	65
2.3.3.1 Harfe .....	65
2.4 Saiteninstrumente mit Bündeln .....	65
2.4.1 Übliche Notation für gebundene Saiteninstrumente .....	65
Referenz für gebundene Saiteninstrumente .....	65
Seitennummerbezeichnung .....	65
Standardtabaturen .....	66
Angepasste Tabaturen .....	67
Bund-Diagramm-Beschriftung .....	68
Fingersatz der rechten Hand .....	69
2.4.2 Gitarre .....	70
Position und Barret anzeigen .....	70
Flageolett und gedämpfte Noten .....	70
2.4.3 Banjo .....	70
Banjo-Tabaturen .....	70
2.5 Schlagzeug .....	71
2.5.1 Übliche Notation für Schlagzeug .....	71
2.5.1.1 Referenz für Schlagzeug .....	71
2.5.1.2 Grundlagen der Schlagzeugnotation .....	71
2.5.1.3 Schlagzeugsysteme .....	71
2.5.1.4 Geisternoten .....	74
2.6 Blasinstrumente .....	75
2.6.1 Übliche Notation für Bläser .....	75
Referenz für Blasinstrumente .....	75
Fingersatz .....	75
2.6.2 Dudelsack .....	75
Dudelsack-Defintionen .....	75
Dudelsack-Beispiele .....	76
2.7 Notation von Akkorden .....	77
2.7.1 Akkord-Modus .....	77
Überblick über den Akkord-Modus .....	77
Übliche Akkorde .....	77
Erweiterte und modifizierte Akkorde .....	80
2.7.2 Akkorde anzeigen .....	80
Akkordbezeichnungen drucken .....	80
Akkordbezeichnungen anpassen .....	83
2.7.3 Generalbass .....	83
Grundlagen des Bezifferten Basses .....	83
Eingabe des Generalbass' .....	86
Generalbass anzeigen .....	86

2.8	Notation von alter Musik	86
2.8.1	Einführung in die Notation Alter Musik	86
2.8.1.1	Notation von Alter Musik, die unterstützt wird	86
2.8.2	Alternative Notenzeichen	86
2.8.2.1	Notenköpfe Alter Musik	86
2.8.2.2	Versetzungszeichen Alter Musik	87
2.8.2.3	Pausen Alter Musik	88
2.8.2.4	Schlüssel Alter Musik	88
2.8.2.5	Fähnchen Alter Musik	90
2.8.2.6	Taktangaben Alter Musik	91
2.8.3	Zusätzliche Notenzeichen	92
2.8.3.1	Artikulationszeichen Alter Musik	92
2.8.3.2	Custodes	93
2.8.3.3	Divisiones	94
2.8.3.4	Ligaturen	94
2.8.3.5	Weißer Mensuralligaturen	95
2.8.3.6	Ligaturen der gregorianischen Quadratnotation	96
2.8.4	Vordefinierte Umgebungen	102
2.8.4.1	Gregorianische Gesangs-Kontexte	102
2.8.4.2	Mensural-Kontexte	102
2.8.5	Transkription Alter Musik	103
2.8.5.1	Alte und moderne Edition aus einer Quelldatei	103
2.8.5.2	Incipite	103
2.8.5.3	Mensurstriche	103
2.8.5.4	Gregorianischen Choral transkribieren	103
2.8.6	Herausgeberische Anmerkungen	103
2.8.6.1	Vorgeschlagene Versetzungszeichen	104
2.8.6.2	Notation barocker Rhythmen	104
2.9	Weltmusik	104
2.9.1	Arabische Musik	104
	References for Arabic music	104
	Arabic note names	104
	Arabic key signatures	104
	Arabic time signatures	104
	Arabic music example	104
	Weitere Literatur	104
<b>3</b>	<b>Allgemeine Eingabe und Ausgabe</b>	<b>105</b>
3.1	Eingabestruktur	105
3.1.1	Struktur einer Partitur	105
3.1.2	Mehrere Partituren in einem Buch	105
3.1.3	Die Dateistruktur	105
3.2	Titel	105
3.2.1	Titel erstellen	105
3.2.2	Eigene Titel	105
3.2.3	Verweis auf die Seitenzahlen	105
3.2.4	Inhaltsverzeichnis	105
3.3	Arbeiten an Eingabe-Dateien	105
3.3.1	LilyPond-Dateien einfügen	105
3.3.2	Verschiedene Editionen aus einer Quelldatei	105
	Variablen benutzen	105
	Marken benutzen	105
3.3.3	Zeichenkodierung	106
3.3.4	LilyPond-Notation anzeigen	106

3.4	Ausgabe kontrollieren .....	106
3.4.1	Notationsfragmente extrahieren .....	106
3.4.2	Korrigierte Musik überspringen .....	106
3.5	MIDI-Ausgabe .....	106
3.5.1	MIDI-Dateien erstellen .....	106
	Instrumentenbezeichnungen .....	106
3.5.2	Der MIDI-Block .....	106
3.5.3	Was geht in die MIDI-Ausgabe .....	106
	In MIDI unterstützt .....	106
	In MIDI nicht unterstützt .....	106
3.5.4	Wiederholungen im MIDI .....	106
3.5.5	MIDI-Lautstärke kontrollieren .....	106
	Dynamik-Zeichen .....	106
	MIDI-Lautstärke .....	106
	Verschiedene Instrumente angleichen (i) .....	106
	Verschiedene Instrumente angleichen (ii) .....	106
<b>4</b>	<b>Abstände .....</b>	<b>107</b>
4.1	Papier und Seiten .....	107
4.1.1	Papierformat .....	107
4.1.2	Seitenformatierung .....	107
4.2	Notenlayout .....	107
4.2.1	Die Notensystemgröße einstellen .....	107
4.2.2	Partiturlayout .....	107
4.3	Umbrüche .....	107
4.3.1	Zeilenumbrüche .....	107
4.3.2	Seitenumbrüche .....	107
4.3.3	Optimale Seitenumbrüche .....	107
4.3.4	Optimale Umbrüche zum Blättern .....	107
4.3.5	Minimale Seitenumbrüche .....	107
4.3.6	Ausdrückliche Umbrüche .....	107
4.3.7	Eine zusätzliche Stimme für Umbrüche benutzen .....	107
4.4	Vertikale Abstände .....	107
4.4.1	Vertikale Abstände innerhalb eines Systems .....	107
4.4.2	Vertikale Abstände zwischen Systemen .....	107
4.4.3	Explizite Positionierung von Systemen .....	107
4.4.4	Vertikale Abstände mit zwei Durchgängen .....	107
4.4.5	Vermeidung von vertikalen Zusammenstößen .....	107
4.5	Horizontale Abstände .....	107
4.5.1	Überblick über horizontale Abstände .....	107
4.5.2	Eine neuer Bereich mit anderen Abständen .....	107
4.5.3	Horizontale Abstände verändern .....	107
4.5.4	Zeilenlänge .....	107
4.5.5	Proportionale Notation .....	108
4.6	Die Musik auf weniger Seiten zwingen .....	108
4.6.1	Abstände anzeigen lassen .....	108
4.6.2	Abstände verändern .....	108

<b>5</b>	<b>Standardeinstellungen verändern</b>	<b>110</b>
5.1	Interpretationsumgebungen	110
5.1.1	Was sind Umgebungen?	110
	Score - Meister aller Kontexte	110
	Oberste Kontexte – Container für Systeme	110
	Mittlere Kontexte – Systeme	110
	Unterste Kontexte – Stimmen	110
5.1.2	Umgebungen erstellen	110
5.1.3	Umgebungs-Plugins verändern	110
5.1.4	Die Standardeinstellungen von Umgebungen ändern	110
5.1.5	Neue Umgebungen definieren	110
5.1.6	Umgebungen aneinander ausrichten	110
5.2	Die Referenz der Programminterna erklärt	110
5.2.1	Zurechtfinden in der Programmreferenz	110
5.2.2	Layout-Schnittstellen	110
5.2.3	Die Grob-Eigenschaften	110
5.2.4	Benennungsübereinkommen	110
5.3	Eigenschaften verändern	110
5.3.1	Überblick über verändernde Eigenschaften	110
5.3.2	Der <code>\set</code> -Befehl	110
5.3.3	Der <code>\override</code> -Befehl	110
5.3.4	<code>\set</code> versus <code>\override</code>	110
5.3.5	Der <code>\tweak</code> -Befehl	110
5.4	Nützliche Konzepte und Eigenschaften	110
5.4.1	Eingabe-Modi	110
5.4.2	Richtung und Platzierung	110
5.4.3	Abstände und Maße	111
5.4.4	Strecker	111
5.5	Übliche Eigenschaften	111
5.5.1	Die Sichtbarkeit von Objekten kontrollieren	111
	Einen stencil entfernen	111
	Objekten unsichtbar machen	111
	Objekte weißmalen	111
	<code>break-visibility</code> (unsichtbar machen) benutzen	111
	Besonderheiten	111
5.5.2	Zeilenstile	111
5.5.3	Drehen von Objekten	111
5.5.4	Umgebungen ausrichten	111
5.6	Fortgeschrittene Optimierungen	111
5.6.1	Vertikale Gruppierung der grafischen Objekte („grob“s)	111
5.6.2	Enden von Streckern verändern	111
5.6.3	stencils verändern	111
5.6.4	Formen verändern	111
5.7	Erklärung von speziellen Optimierungen	111
5.7.1	Alte Kontexte	111



<b>6</b>	<b>Schnittstellen für Programmierer</b>	<b>112</b>
6.1	Musikalische Funktionen	112
6.1.1	Überblick über musikalische Funktionen	112
6.1.2	Einfache Ersetzungsfunktionen	112
6.1.3	Paarige Ersetzungsfunktionen	112
6.1.4	Mathematik in Funktionen	112
6.1.5	Leere Funktionen	112
6.1.6	Funktionen ohne Argumente	112
6.1.7	Überblick über vorhandene musikalische Funktionen	112
6.2	Schnittstelle für Programmierer	115
6.2.1	Eingabevariablen und Scheme	115
6.2.2	Interne Repräsentation der Musik	115
6.3	Komplizierte Funktionen erstellen	115
6.3.1	Musikalische Funktionen darstellen	115
6.3.2	Eigenschaften von Musikobjekten	115
6.3.3	Verdoppelung einer Note mit Bindebögen (Beispiel)	115
6.3.4	Artikulationszeichen zu Noten hinzufügen (Beispiel)	115
6.4	Programmierungsschnittstelle für Textbeschriftungen	115
6.4.1	Beschriftungskonstruktionen in Scheme	115
6.4.2	Wie Beschriftungen intern funktionieren	115
6.4.3	Neue Definitionen von Beschriftungsbefehlen	115
6.4.4	Neue Definitionen von Beschriftungsbefehlen für Listen	115
6.5	Kontexte für Programmierer	115
6.5.1	Kontextauswertung	115
6.5.2	Eine Funktion auf alle Layout-Objekte anwenden	115
6.6	Scheme-Vorgänge als Eigenschaften	115
6.7	TODO moved into scheme	115
6.7.1	Scheme-Code anstelle von <code>weak</code> verwenden	115
6.7.2	Schwierige Korrekturen	115
<b>Anhang A</b>	<b>Literatur</b>	<b>116</b>
<b>Anhang B</b>	<b>Notationsübersicht</b>	<b>117</b>
B.1	Liste der Akkordbezeichnungen	117
B.2	MIDI-Instrumente	118
B.3	Liste der Farben	119
B.4	Die Feta-Schriftart	121
B.5	Notenkopfstile	134
B.6	Text markup commands	135
B.6.1	Font	135
B.6.2	Align	144
B.6.3	Graphic	157
B.6.4	Music	161
B.6.5	Instrument Specific Markup	165
B.6.6	Other	167
B.7	Text markup list commands	171
B.8	Liste der Artikulationszeichen	172
B.9	Alle Kontexteigenschaften	173
B.10	Eigenschaften des Layouts	183
B.11	Bezeichner	196
B.12	Scheme-Funktionen	199

Anhang C	Befehlsübersicht .....	<b>218</b>
Anhang D	GNU Free Documentation License.....	<b>222</b>
Anhang E	Index der LilyPond-Befehle .....	<b>228</b>
Anhang F	LilyPond-Index .....	<b>234</b>

# 1 Musikalische Notation

This chapter explains how to create musical notation.

## 1.1 Tonhöhen



This section discusses how to specify the pitch of notes.

### 1.1.1 Tonhöhen setzen

Into text.

#### Absolute Oktavenbezeichnung

Tonhöhenbezeichnungen werden durch Kleinbuchstaben von a bis g angegeben.<sup>1</sup> Eine aufsteigende C-Dur-Tonleiter wird wie folgt notiert:

```
\clef bass
c d e f g a b c'
```



Die Notenbezeichnung c schreibt als Note ein kleines C, eine Oktave unter dem eingestrichenen C.

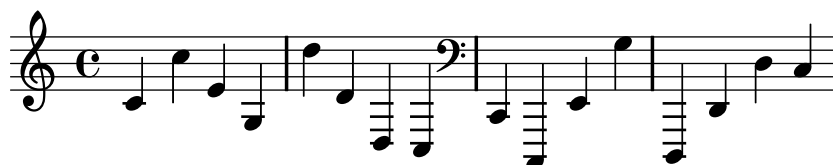
```
\clef treble
c1
\clef bass
c1
```

<sup>1</sup> Die Benutzung deutscher Notenbezeichnungen mit der Unterscheidung von b und h ist auch möglich, siehe [\[Notenbezeichnungen in anderen Sprachen\]](#), Seite 5.



Zusätzliche Oktavbestimmung wird mit einer Anzahl von Apostrophen (',' ) oder Kommas (',') vorgenommen. Jeder Apostroph erhöht die Note um eine Oktave, jedes Komma erniedrigt sie um eine Oktave.

```
\clef treble
c' c'' e' g d'' d' d c
\clef bass
c, c,, e, g d,, d, d c
```



Eine alternative Methode gibt am Anfang die Oktave vor, innerhalb derer die Noten gesetzt werden, dabei werden unter Umständen weniger Oktavangaben ( ' oder , ) benötigt. Siehe auch [\[Relative Oktavenbezeichnung\]](#), Seite 2.

## Relative Oktavenbezeichnung

Oktaven werden angegeben, indem man ' oder , an die Notenbezeichnung hängt. Wenn Sie schon existierende Musik kopieren, passiert es schnell, eine Note aus Versehen in die falsche Oktave zu setzen, und der Fehler ist schwer zu finden. Der relative Oktaven-Modus verhindert solche Fehler, indem mögliche Fehler stark vergrößert werden: ein einziger Oktavierungsfehler wirkt sich auf den gesamten Rest des Stückes aus.

Die Syntax des Befehls lautet:

```
\relative Referenzoktave musikalischer Ausdruck
```

oder:

```
\relative musikalischer Ausdruck
```

Das eingestrichene C (c') wird als Referenzoktave angenommen, wenn sie nicht extra angegeben wird.

Die Oktave von Noten, die im musikalischen Ausdruck notiert sind, wird wie folgt erschlossen: Wenn keine Oktavversetzungszeichen benutzt werden, wird als Intervall zwischen der Noten und der vorhergehenden immer eine Quarte oder kleiner angenommen. Dieser Abstand wird ohne Rücksicht auf Alterationen bestimmt. Eine übermäßige Quarte ist also ein kleineres Intervall als eine verminderte Quinte, auch wenn beide sechs Halbtöne groß sind.

Die Oktavversetzungszeichen ' und , können hinzugefügt werden, um die Tonhöhe um eine Oktave zu erhöhen oder zu erniedrigen. Wenn der relative Modus beginnt, kann ein Referenzton angegeben werden, der als die vorhergehende Note für die erste Tonhöhe des musikalischen Ausdrucks verwendet wird. Wenn dieser Referenzton nicht angegeben wird, wird das eingestrichene C verwendet.

So funktioniert der relative Modus:

```
\relative c'' {
  b c d c b c bes a
}
```



Oktavversetzungen müssen für alle Intervalle angezeigt werden, die größer als eine Quarte sind.

```
\relative c'' {
  c g c f, c' a, e''
}
```



Wenn der vorherige Ausdruck ein Akkord ist, wird die erste Note des Akkordes benutzt, um die erste Note des nächsten Akkordes zu bestimmen.

```
\relative c' {
  c <c e g>
  <c' e g>
  <c, e' g>
}
```



Die Tonhöhe nach `\relative` muss eine Notenbezeichnung enthalten.

Die relative Veränderung wirkt sich nicht auf Transposition (`\transpose`), Akkordnotation (`\chordmode`) oder `\relative`-Abschnitte aus. Um den relativen Modus innerhalb von transponierter Musik zu verwenden, muss ein zusätzliches `\relative` innerhalb der Klammern des `\transpose`-Befehls gesetzt werden.

## Versetzungszeichen

Ein Kreuz wird eingegeben, indem man `-is` an die Notenbezeichnung hängt, ein b durch `-es`. Doppelkreuze und Doppel-Bs werden durch Hinzufügen von `-isis` und `-eses` hinter die Notenbezeichnung erzeugt.

```
a2 ais a aes
a2 aisis a aeses
```



Ein Auflösungszeichen macht die Wirkung eines Kreuzes oder Bs rückgängig. Diese Auflösungszeichen werden jedoch nicht als Suffix einer Tonhöhenbezeichnung eingegeben, sondern sie ergeben sich (automatisch) aus dem Kontext, wenn die nicht alterierte Notenbezeichnung eingegeben wird.

```
a4 aes a2
```



Die Sequenz `d e f` wird interpretiert als: „Setze eine D-Noten, eine E-Note und eine F-Note,“ unabhängig von den Vorzeichen der Tonart. Mehr Information über den Unterschied zwischen musikalischem Inhalt und der Präsentation dieses Inhalts siehe [Abschnitt “Accidentals and key signatures” in Handbuch zum Lernen](#).

```
\key d \major
d e f g
d e fis g
```



Versetzungszeichen für Vierteltöne werden durch Anhängen der Endungen `-eh` (Erniedrigung) und `-ih` (Erhöhung) an den Tonhöhenbuchstaben erstellt. Das Beispiel zeigt eine in Vierteltönen aufsteigende Serie vom kleinen C.

```
ceseh1 ces ceh c cih cis csih
```



Mikrotöne werden auch in die MIDI-Dateien geschrieben.

Normalerweise werden Versetzungszeichen automatisch gesetzt, aber sie können auch manuell hinzugefügt werden. Ein erinnerndes Versetzungszeichen kann erzwungen werden, indem man ein Ausrufungszeichen (!) hinter die Notenbezeichnung schreibt. Ein warnendes Versetzungszeichen (also ein Vorzeichen in Klammern) wird durch Anfügen eines Fragezeichens (?) erstellt. Mit diesen zusätzlichen Zeichen kann man sich auch Auflösungszeichen ausgeben lassen.

```
cis cis cis! cis? c c? c! c
```



## Übliche Veränderungen der Einstellungen

Den Satzregeln für den Notensatz folgend wird ein Auflösungszeichen dann ausgegeben, wenn ein vorhergehendes Versetzungszeichen rückgängig gemacht werden soll. Um dieses Verhalten zu ändern, muss `\set Staff.extraNatural = ##f` eingesetzt werden.

```
ceses4 ces cis c
\set Staff.extraNatural = ##f
ceses4 ces cis c
```



## Siehe auch

Die automatische Setzung von Versetzungszeichen kann auf viele Arten beeinflusst werden. Mehr Information dazu siehe [\[Automatische Versetzungszeichen\]](#), Seite 10.

Programmreferenz: [Abschnitt "LedgerLineSpanner"](#) in *Programmreferenz*, [Abschnitt "Note-Head"](#) in *Programmreferenz*.

## Bekannte Probleme und Warnungen

Es gibt keine allgemein anerkannten Standards für die Notation von Dreiviertelton-Erniedrigungszeichen. LilyPonds Symbol entspricht also keinem Standard.

## Notenbezeichnungen in anderen Sprachen

Es gibt vordefinierte Bezeichnungen für die Notenbezeichnungen in anderen Sprachen als Englisch. Um sie zu benutzen, muss nur die entsprechende Datei für die jeweilige Sprache eingefügt werden. Zum Beispiel fügt man mit `\include "deutsch.ly"` die Notendefinitionen für die deutsche Sprache am Anfang der Datei hinzu. In der Tabelle sind die existierenden Definitionen mit den dazugehörigen Notenbezeichnungen dargestellt.

	Notenbezeichnungen								Kreuz	B	Doppelkreuz	Dopp
nederlands.ly	c	d	e	f	g	a	bes	b	-is	-es	-isis	-es
english.ly	c	d	e	f	g	a	bf	b	-s/-sharp	-f/-flat	-ss/-x/	-ff/
											-sharpsharp	-fl
deutsch.ly	c	d	e	f	g	a	b	h	-is	-es	-isis	-es
norsk.ly	c	d	e	f	g	a	b	h	-iss/-is	-ess/-es	-ississ/-isis	-es
svenska.ly	c	d	e	f	g	a	b	h	-iss	-ess	-ississ	-es
italiano.ly	do	re	mi	fa	sol	la	sib	si	-d	-b	-dd	-bb
catalan.ly	do	re	mi	fa	sol	la	sib	si	-d/-s	-b	-dd/-ss	-bb
espanol.ly	do	re	mi	fa	sol	la	sib	si	-s	-b	-ss	-bb

Auf Holländisch, Deutsch, Norwegisch und Schwedisch (u. a.) werden die Erniedrigungen von ‚a‘ wie `aes` und `aeses` zu `as` und `ases` (oder auch `asas`) zusammengezogen. In manchen Sprachen sind nur diese Kurzformen definiert (das gilt auch für die Endungen der Vierteltöne).

Bestimmte Musik verwendet Alterationen, die Bruchteile von den „üblichen“ Kreuzen oder Bs sind. Die Notenbezeichnungen für Vierteltöne für die verschiedenen Sprachen sind in der folgenden Tabelle aufgeführt. Die Präfixe „Semi-“ und „Sesqui-“ bedeuten „halb“ bzw. „eineinhalb“. Für Norwegisch, Schwedisch, Katalanisch und Spanisch sind noch keine eigenen Namen definiert.

	Notenbezeichnungen								Semi- kreuz	Semi- B	Sesqui- Kreuz	Sesqui- B
nederlands.ly	c	d	e	f	g	a	bes	b	-ih	-eh	-isih	-eseh
english.ly	c	d	e	f	g	a	bf	b	-qs	-qf	-tqs	-tqf
deutsch.ly	c	d	e	f	g	a	b	h	-ih	-eh	-isih	-eseh
norsk.ly	c	d	e	f	g	a	b	h				
svenska.ly	c	d	e	f	g	a	b	h				
italiano.ly	do	re	mi	fa	sol	la	sib	si	-sd	-sb	-dsd	-bsb
catalan.ly	do	re	mi	fa	sol	la	sib	si				
espanol.ly	do	re	mi	fa	sol	la	sib	si				

### 1.1.2 Viele Tonhöhen gleichzeitig verändern

#### Oktavenüberprüfung

Durch Oktavenüberprüfung können Fehler einfacher entdeckt werden: nach einer Note kann =*Apostrophe* geschrieben werden, womit angezeigt wird, was ihre wirkliche Oktave sein soll. Im folgenden Beispiel

```
\relative c' { c=' b=' d,=' }
```

erzeugt das `d` eine Warnung, weil ein `d'` erwartet wird (denn zwischen `b'` und `d'` befindet sich nur eine Terz), aber ein `d'` ist notiert. In der Notenausgabe wird die Oktave zu `d'` korrigiert und die nächste Note wird relativ zu `d'` anstelle von `d` errechnet.

Es gibt auch eine Oktavenüberprüfung, die keine sichtbare Ausgabe erzeugt. Die Syntax:

```
\octaveCheck Tonhöhe
```

Hierdurch wird überprüft, dass die *Tonhöhe* (ohne Apostroph) der *Tonhöhe* (mit Apostroph) entspricht. Wenn sie sich nicht entsprechen, wird eine Warnung ausgegeben und die Oktave wird korrigiert. Die *Tonhöhe* wird nicht als Note gesetzt.

Im nächsten Beispiel erzeugt die erste Überprüfung keine Warnung, weil das `e` (im relativen Modus) innerhalb einer Quarte zum `a'` liegt. Die zweite Überprüfung aber erzeugt eine Warnung, weil das `e` mehr als eine Quarte vom `b'` entfernt ist. Die Warnung wird ausgegeben und die Oktave wird korrigiert, so dass auch die folgenden Noten wieder in der richtigen Oktave gesetzt werden.

```
\relative c' {
  e
  \octaveCheck a'
  \octaveCheck b'
}
```

Die Oktave einer Note, der eine Oktavüberprüfung angefügt wurde, wird in Hinsicht auf die vorherige Note bestimmt. Im nächsten Fragment ist die letzte Note ein `a'`. Die Oktavenüberprüfung stellt fest, dass zwischen `e'` und kleinem `b` eine Quarte Abstand ist und dass die folgende Note, ein `a'`, sich wieder innerhalb einer Quarte vom `e'` aus befindet. Die Überprüfung gibt also einen Erfolgswert zurück und die Notenausgabe wird nicht verändert.

```
\relative c' {
  e
  \octaveCheck b
  a
}
```





## Transposition

Ein musikalischer Ausdruck kann mit dem Befehl `\transpose` transponiert werden. Die Syntax lautet:

```
\transpose von nach mus. Ausdruck
```

Das bedeutet, dass der *mus. Ausdruck* um das Intervall zwischen den Tonhöhen *von* und *nach* transponiert wird: Jede Note, die die Tonhöhe *von* hat, wird in die Tonhöhe *nach* umgewandelt.

So kann z. B. ein Stück in D-Dur, wenn es für den Sänger etwas zu tief ist, mit dem Befehl

```
\transpose d e ...
```

nach E-Dur transponiert werden.

Oder eine Violinstimme, die so notiert wird, wie sie erklingt, soll von einer A-Klarinette gespielt werden. Hier ist ein klingendes A als C notiert, so dass alles also eine kleine Terz tiefer erklingt, als es notiert ist. Für die Erzeugung der Klarinettenstimme muss folgender Befehl verwendet werden:

```
\transpose a c ...
```

`\transpose` unterscheidet enharmonische Verwechslungen: sowohl `\transpose c cis` als auch `\transpose c des` transponieren die Musik einen Halbton nach oben. Aber die erste Version gibt als Versetzungszeichen Kreuze aus, die zweite dagegen B-Versetzungszeichen.

```
mus = { \key d \major cis d fis g }
\new Staff {
  \clef "F" \mus
  \clef "G"
  \transpose c g' \mus
  \transpose c f' \mus
}
```



`\transpose` kann auch benutzt werden, um die geschriebenen Noten eines transponierenden Instruments zu notieren. Tonhöhen in LilyPond werden üblicherweise notiert, wie sie erklingen, aber man kann auch eine andere Tonart verwenden. Noten einer B-Trompete, die mit einem klingenden D anfangen, könnte man also auch so eingeben:

```
\transpose c bes { e4 ... }
```

Um die Noten dann wiederum als Trompetenstimme zu drucken (also einen Ganzton tiefer, als sie erklingen), setzt man einfach um sie herum eine weitere Transposition:

```
\transpose bes c { \transpose c bes { e4 ... } }
```

## Siehe auch

Programmreferenz: [Abschnitt "TransposedMusic"](#) in *Programmreferenz*.

Beispiel: [Abschnitt "Pitches"](#) in *Beispiele*.

## Bekannte Probleme und Warnungen

Wenn Sie sowohl `\transpose` als auch `\relative` benutzen wollen, muss die `\transpose`-Umgebung sich außerhalb der `\relative`-Umgebung befinden, da `\relative` keine Auswirkungen auf Noten hat, die sich innerhalb von `\transpose` befinden.

### 1.1.3 Tonhöhen anzeigen lassen

#### Notenschlüssel

Der Schlüssel zeigt eine bestimmte Systemlinie an und markiert die Tonhöhe, mit der sie korrespondiert. Ein Schlüssel wird mit dem `\clef`-Befehl gesetzt.

```
{ c'2 \clef alto g'2 }
```



Unterstützt sind folgende Schlüssel:

Schlüssel	Lage
<code>treble</code> (Violinschlüssel)	G-Schlüssel auf der zweiten Linie
<code>alto</code> , <code>C</code> (Bratschenschlüssel)	C-Schlüssel auf der dritten Linie
<code>tenor</code> (Tenorschlüssel)	C-Schlüssel auf der vierten Linie
<code>bass</code> , <code>F</code> (Bassschlüssel)	F-Schlüssel auf der vierten Linie
<code>french</code> (Franz. Violinschlüssel)	G-Schlüssel auf der ersten Linie
<code>soprano</code> (Sopranschlüssel)	C-Schlüssel auf der ersten Linie
<code>mezzosoprano</code> (Mezzosopranschlüssel)	C-Schlüssel auf der zweiten Linie
<code>baritone</code> (Baritonschlüssel)	C-Schlüssel auf der fünften Linie
<code>varbaritone</code> (Bariton-F-Schlüssel)	F-Schlüssel auf der dritten Linie
<code>subbass</code> (Kontrabassschlüssel)	F-Schlüssel auf der fünften Linie
<code>percussion</code>	Schlagzeugschlüssel
<code>tab</code>	Tabulaturschlüssel

Indem `_8` oder `^8` an die jeweilige Schlüsselbezeichnung angehängt wird, wird der Schlüssel um eine Oktave nach oben oder unten transponiert, mit `_15` oder `^15` um zwei Oktaven. Die Schlüsselbezeichnung muss in Anführungszeichen gesetzt werden, wenn sie Unterstriche oder Zahlen enthält, siehe Beispiel:

```
\clef "G_8" c4
```



## Übliche Veränderungen der Einstellungen

Dem Befehl `\clef "treble_8"` entspricht die entsprechende separate Einstellung von `clefGlyph`, `clefPosition` (womit y- und x-Position des Schlüssel bestimmt werden), `middleCPosition` und `clefOctavation`. Ein Schlüssel wird immer dann ausgegeben, wenn eine dieser Eigenschaften sich ändert. Im nächsten Beispiel werden Möglichkeiten gezeigt, die Eigenschaften manuell zu setzen.

```
{
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'4
  \set Staff.clefGlyph = #"clefs.G"
  c'4
  \set Staff.clefGlyph = #"clefs.C"
  c'4
  \set Staff.clefOctavation = #7
  c'4
  \set Staff.clefOctavation = #0
  \set Staff.clefPosition = #0
  c'4
  \clef "bass"
  c'4
  \set Staff.middleCPosition = #4
  c'4
}
```



## Siehe auch

Handbuch: [Abschnitt 1.2.6.1 \[Verzierungen\]](#), Seite 26.

Programmreferenz: [Abschnitt “Clef” in Programmreferenz](#).

## Tonartbezeichnung

Die Vorzeichen zeigen die Tonart an, in welcher ein Stück notiert ist. Es handelt sich um eine Anzahl von Alterationszeichen (Kreuzen oder Bs) am Beginn jedes Notensystems.

Das Setzen und Ändern von Tonarteinstellungen wird mit dem `\key`-Befehl vorgenommen.

`\key Tonhöhe Art`

Der Wert *Art* sollte entweder `\major` oder `\minor` sein, um Moll oder Dur der *Tonhöhe* zu erhalten. Es können auch Modusbezeichnungen für Kirchentonarten verwendet werden: `\ionian` (Ionisch), `\locrian` (Locrisch), `\aeolian` (Aeolisch), `\mixolydian` (Mixolydisch), `\lydian` (Lydisch), `\phrygian` (Phrygisch) und `\dorian` (Dorisch).

Dieser Befehl ändert die Kontexteigenschaft `Staff.keySignature`. Vorzeichen, die nicht dem Standard entsprechen, können manuell mit dieser Eigenschaft eingegeben werden.

Versetzungszeichen und Vorzeichen können am Anfang etwas verwirrend sein, da unveränderte Noten je nach Tonart ein Auflösungszeichen bekommen können (Beispiel). Mehr Information in den Abschnitten [\[Versetzungszeichen\]](#), Seite 3 oder [Abschnitt “Accidentals and key signatures” in Handbuch zum Lernen](#).

```
\key g \major
f1
fis
```



## 1.2 Rhythmus



This section discusses rhythms, durations, and bars.

### 1.2.1 Rhythmen eingeben

#### 1.2.1.1 Tondauern

Im Noten-, Akkord- und Liedtextmodus werden Notenlängen (Dauern) durch Zahlen und Punkte notiert: Dauern werden als reziproke Werte geschrieben. Zum Beispiel wird eine Viertelnote mit 4 notiert (weil sie eine 1/4-Note ist), eine halbe Note mit 2 (weil sie eine 1/2-Note ist). Noten, die länger als eine Ganze sind, müssen mit `\longa` (für die Longa, also vier Ganze) und `\breve` (für die Brevis, auch Doppelganze genannt) notiert werden.

```
c'\breve
c'1 c'2 c'4 c'8 c'16 c'32 c'64 c'64
r\longa r\breve
r1 r2 r4 r8 r16 r32 r64 r64
```



Wenn die Dauer hinter einer Notenbezeichnung nicht angegeben ist, wird die Dauer der vorhergehenden Note eingesetzt. Der Standardwert für die erste Note ist eine Viertel.

```
{ a a a2 a a4 a a1 a }
```



Um punktierte Notendauern zu erhalten, muss einfach nur ein Punkt (.) hinter die Zahl der Dauer gesetzt werden. Zwei Punkte ergeben eine doppelte Punktierung.

a'4 b' c''4. b'8 a'4. b'4.. c''8.



## Vordefinierte Befehle

Punkte werden normalerweise nach oben verschoben, damit sie die Notenlinien nicht berühren. Das gilt aber nicht für mehrstimmige Passagen. Mit den folgenden Befehlen kann ein anderes Verhalten der Punktierung erreicht werden.

`\dotsUp` (Der Punkt wird nach oben verschoben.), `\dotsDown` (Der Punkt wird nach unten verschoben.), `\dotsNeutral`.

## Siehe auch

Programmreferenz: [Abschnitt “Dots” in Programmreferenz](#), and [Abschnitt “DotColumn” in Programmreferenz](#).

### 1.2.1.2 Andere rhythmische Aufteilungen

Triolen und andere rhythmische Aufteilungen werden aus einem musikalischen Ausdruck erstellt, indem dessen Tondauern mit einem Bruch multipliziert werden.

`\times Bruch musikalischer Ausdruck`

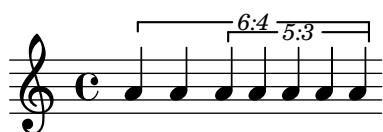
Die Dauer eines *musikalischen Ausdrucks* wird mit dem Bruch multipliziert. Der Nenner des Bruchs wird über den Noten ausgegeben, optional mit einer eckigen Klammer, die die Noten einfasst. Die üblichste Aufteilung ist die Triole, in welcher drei Noten die Länge von zwei haben, der Wert jeder einzelnen Note ist also 2/3 der notierten Länge.

`g'4 \times 2/3 {c'4 c' c'} d'4 d'4`



Diese Brüche können auch ineinander geschachtelt werden, etwa so:

```
\override TupletNumber #'text = #tuplet-number::calc-fraction-text
\times 4/6 {
  a4 a
  \times 3/5 { a a a a a }
}
```



## Vordefinierte Befehle

`\tupletUp`, `\tupletDown`, `\tupletNeutral`.

## Übliche Veränderungen der Einstellungen

Der Wert von `tupletSpannerDuration` definiert, wie lange jede Klammer dauert. Mit entsprechender Einstellung kann man beliebig viele Triolen schreiben, aber nur einmal den Befehl `\times` benutzen. Im nächsten Beispiel etwa werden zwei Triolen gedruckt, `\times` aber nur einmal benutzt. Das gilt natürlich auch für alle anderen Brüche.

```
\set tupletSpannerDuration = #(ly:make-moment 1 4)
\times 2/3 { c8 c c c c c }
```



Mehr Information zu `make-moment` findet sich im Abschnitt [Abschnitt 1.2.6.3 \[Verwaltung der Zeiteinheiten\]](#), Seite 29.

Die Formatierung der Nummer wird durch die Eigenschaft `text` in `TupletNumber` bestimmt. Die Standardeinstellung gibt nur den Nenner aus, aber wenn `TupletNumber` auf den Wert `tuplet-number::calc-fraction-text` gestellt wird, wird *Zähler:Nenner* ausgegeben.

Um gar keine Nummern anzeigen zu lassen, kann folgender Code benutzt werden:

```
\times 2/3 { c8 c c } \times 2/3 { c8 c c }
\override TupletNumber #'transparent = ##t
\times 2/3 { c8 c c } \times 2/3 { c8 c c }
```



Mit der `\tweak`-Funktion können die Zahlen von geschachtelten Brüchen eingestellt werden, die zur gleichen Zeit beginnen. Im Beispiel unten wird mit `\tweak` definiert, dass für die äußere Klammer der Bruch ausgegeben wird, für die innere aber nur der Nenner.

```
\new Staff {
  \tweak #'text #tuplet-number::calc-fraction-text
  \times 4/3 {
    \tweak #'text #tuplet-number::calc-denominator-text
    \times 2/3 { c'8[ c'8 c'8] }
    \times 2/3 { c'8[ c'8 c'8] }
    \times 2/3 { c'8[ c'8 c'8] }
  }
}
```



Im nächsten Beispiel werden `\tweak` und `\override` zusammen verwendet, um die Darstellung der Klammer (`TupletBracket`) zu bestimmen. Mit dem ersten `\tweak` wird die Klammer der äußeren Triole über dem Notensystem platziert. Das zweite `\tweak` platziert die erste der drei inneren Klammern unter dem System (von der Klammer wird hier nur der Nenner des Bruchs dargestellt). Dieses Paar von `\tweak`-Einstellungen wirkt sich jedoch nur auf die äußere und die *erste* innere Klammer aus, weil nur sie zur gleichen Zeit beginnen. Mit dem `\override`-Befehl kann die Richtung der zweiten und dritten `TupletBracket` verändert werden, in diesem Fall wird sie unter das Notensystem geschrieben.

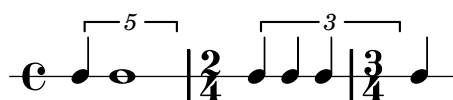
```

\new Staff {
  \tweak #'text #tuplet-number::calc-fraction-text
  \tweak #'direction #up
  \times 4/3 {
    \tweak #'direction #down
    \times 2/3 { c'8[ c'8 c'8] }
    \override TupletBracket #'direction = #down
    \times 2/3 { c'8[ c'8 c'8] }
    \times 2/3 { c'8[ c'8 c'8] }
  }
}

```



Die Klammern können so eingestellt werden, dass sie bis zu einem Taktvorspann oder bis zur nächsten Note reichen.



## Siehe auch

Programmreferenz: [Abschnitt “TupletBracket”](#) in *Programmreferenz*, [Abschnitt “TupletNumber”](#) in *Programmreferenz* und [Abschnitt “TimeScaledMusic”](#) in *Programmreferenz*.

### 1.2.1.3 Tondauern skalieren

Die Dauer von Längen kann mit einem Bruch multipliziert werden, indem hinter die Note „\*N/M“ (oder „\*N“ wenn  $M=1$ ) geschrieben wird. Das beeinflusst nicht die Erscheinung der Note oder Pause im Druckbild. Die Werte können auch kombiniert werden, etwa „\*M\*N“.

Im nächsten Beispiel nehmen die drei ersten Noten genau zwei Schläge ein, aber es wird keine Triolenklammer über ihnen ausgegeben.

```

\time 2/4
a4*2/3 gis4*2/3 a4*2/3
a4 a4 a4*2
b16*4 c4

```



## Siehe auch

Abschnitt [Abschnitt 1.2.1.2 \[Andere rhythmische Aufteilungen\]](#), Seite 12



### 1.2.1.4 Bindebögen

Ein Bindebogen verbindet zwei benachbarte Noten der selben Tonhöhe. Als Resultat wird die Dauer der Notenlänge verlängert. Bindebögen dürfen nicht mit Legatobögen verwechselt werden, durch die die Vortragsart bezeichnet wird, noch mit Phrasierungsbögen, die musikalische Phrasen anzeigen. Eine Bindebogen wird mit der Tilde `~` (AltGr++) notiert.

`e' ~ e' <c' e' g'> ~ <c' e' g'>`



Wenn ein Bindebogen an einen Akkord gehängt wird, werden alle Noten dieses Akkordes übergebunden. Wenn kein Notenkopf passt, wird auch kein Bogen erzeugt. Noten in Akkorden können auch einzeln übergebunden werden, indem sie innerhalb des Akkordes hinter die entsprechende Note geschrieben werden.

`<c~ e g~ b> <c e g b>`



Ein Bindebogen ist nur eine andere Art, die Notendauer zu verlängern, ähnlich wie die Punktierung. Im nächsten Beispiel sind zwei Arten gezeigt, die gleiche Notenlänge zu notieren:



Bindebögen werden verwendet, wenn die Note entweder über die Taktgrenze hinausragt, oder wenn Punktierung nicht benutzt werden kann, um die Verlängerung anzuzeigen. Wenn man Überbindungen verwendet, sollten größere Notenwerte an die Unterteilungen des Taktes angepasst werden.



Wenn sehr viele Noten über die Taktgrenzen hinüber angebunden werden müssen, ist es oft einfacher, die automatische Aufteilung von Noten einzusetzen (siehe [Abschnitt 1.2.3.5 \[Automatische Aufteilung von Noten\]](#), Seite 21). Mit dieser Funktion werden automatisch lange Noten aufgeteilt und über die Taktgrenze übergebunden.

Wenn die zweite Variante einer Wiederholung mit einer Überbindung anfängt, muss der Bindebogen wiederholt werden. Dass geschieht durch den Befehl `\repeatTie`.



## Übliche Veränderungen der Einstellungen

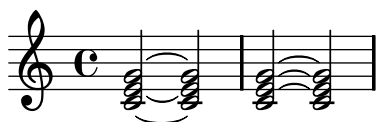
Bindebögen werden teilweise verwendet, um Arpeggien auszuschreiben. In diesem Fall müssen mehrere übergebundene Noten nacheinander erscheinen. Das ist möglich, indem die `tieWaitForNote`-Eigenschaft auf wahr (`##t`) gesetzt wird. Diese Funktion ist auch nützlich, um ein Tremolo an einen Akkord zu binden. Siehe das Beispiel:

```
\set tieWaitForNote = ##t
\grace { c16[~ e~ g]~ } <c, e g>2
\repeat tremolo 8 { c32~ c'~ } <c c,>1
e8~ c~ a~ f~ <e' c a f>2
```



Bindebögen können manuell gesetzt werden, indem die `tie-configuration`-Eigenschaft verändert wird. Die erste Zahl zeigt den Abstand von der Mitte des Notensystems in Notenlinienzwischenräumen, die zweite die Richtung (1=nach oben, -1=nach unten).

```
<c e g>2~ <c e g> |
\override TieColumn #'tie-configuration =
  #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
<c e g>~ <c e g> |
```



## Vordefinierte Befehle

`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieSolid`.

## Siehe auch

Im Handbuch: [Abschnitt 1.2.3.5 \[Automatische Aufteilung von Noten\]](#), Seite 21.

Programmreferenz: [Abschnitt "Tie" in Programmreferenz](#).

## Bekannte Probleme und Warnungen

Der Wechsel zwischen Systemen bei aktiver Überbindung produziert keinen gekrümmten Bogen.

Änderung von Schlüssel oder Oktavierung zwischen übergebundenen Noten ist nicht richtig definiert. In diesen Fällen kann es besser sein, einen Legatobogen zu verwenden.

### 1.2.2 Pausen eingeben

### 1.2.2.1 Pausen

Pausen werden wie Noten eingegeben, ihre Bezeichnung ist `r`.

`r1 r2 r4 r8`



Pausen, die ganze Takte ausfüllen und in der Taktmitte zentriert werden sollen, müssen als mehrtaktige Pausen eingegeben werden. Sie können sowohl für einen einzigen Takt als auch für mehrere Takte verwendet werden, Näheres im Abschnitt [Abschnitt 1.2.2.3 \[Ganztaktige Pausen\]](#), [Seite 18](#).

Um die vertikale Position einer Pause explizit festzulegen, kann eine Note eingegeben werden, gefolgt vom Befehl `\rest`. Die Pause wird dann an die Stelle gesetzt, wo sich sonst die Note befinden würde.

`a'4\rest d'4\rest`



Damit wird die manuelle Formatierung von mehrstimmiger Musik sehr viel einfacher, da die Formatierungsfunktion zur automatischen Auflösung von Zusammenstößen diese Pausen nicht mit einbezieht.

### Siehe auch

Programmreferenz: [Abschnitt "Rest" in Programmreferenz](#).

### 1.2.2.2 Unsichtbare Pausen

Eine unsichtbare Pause (auch als „skip“ oder Übersprungung bezeichnet) kann wie eine Note eingegeben werden, die Notationsbezeichnung ist `s`. Man kann aber auch die Dauer extra angeben mit `\skip Dauer`.

`a4 a4 s4 a4 \skip 1 a4`



Die `s`-Syntax steht nur im Noten- oder Akkordmodus zur Verfügung. In anderen Situationen, z. B. innerhalb eines Liedtextes, muss `\skip` benutzt werden.

```
<<
  \relative { a'2 a2 }
  \new Lyrics \lyricmode { \skip 2 bla2 }
>>
```



bla

Der Übersprungbefehl (`\skip`) ist einfach ein leerer Platzhalter. Durch ihn wird überhaupt nichts gesetzt, auch keine transparenten Objekte.

Die Übersprungung mit `s` hingegen erstellt **Abschnitt “Staff”** in *Programmreferenz* und **Abschnitt “Voice”** in *Programmreferenz* wenn es erforderlich ist, genauso wie Noten und Pausen. Das folgende Beispiel etwa setzt ein leeres Notensystem:

```
{ s4 }
```



Das Fragment `{ \skip 4 }` würde nur eine leere Seite produzieren.

## Siehe auch

Programmreferenz: **Abschnitt “SkipMusic”** in *Programmreferenz*.

### 1.2.2.3 Ganztaktige Pausen

## 1.2.3 Rhythmen anzeigen lassen

### 1.2.3.1 Taktangabe

Taktangaben zeigen das Metrum eines Stückes an: eine regelmäßige Folge von betonten und unbetonten Zeiten. Es wird angezeigt als ein Bruch vor Beginn der Noten.

Die Taktangabe wird mit dem `\time`-Befehl gesetzt.

```
\time 2/4 c'2 \time 3/4 c'2.
```



## Übliche Veränderungen der Einstellungen

Das Symbol, das angezeigt wird, kann durch die `style`-Eigenschaft angepasst werden. Wenn man es auf den Wert `#'()` setzt, wird auch für 4/4- und 2/2-Takte ein Bruch angezeigt.

```
\time 4/4 c'1
\time 2/2 c'1
\override Staff.TimeSignature #'style = #'()
\time 4/4 c'1
\time 2/2 c'1
```



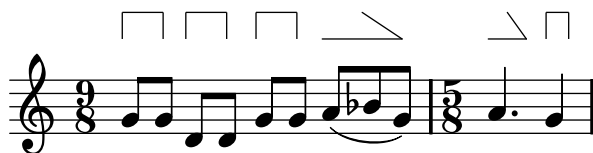
Es gibt noch sehr viel mehr Optionen für das Layout. Siehe den Abschnitt **Abschnitt 2.8.2.6 [Taktangaben Alter Musik]**, Seite 91 für weitere Beispiele.

Der Befehl `\time` stellt die Eigenschaften `timeSignatureFraction`, `beatLength` und `measureLength` im `Timing`-Kontext ein, der normalerweise zu den Eigenschaften aller System

(Abschnitt “Score” in *Programmreferenz*-Ebene) gehört. Die Eigenschaft `measureLength` bestimmt, wo Taktlinien eingefügt werden, und wie automatische Balken gesetzt werden. Ein Verändern der Eigenschaften von `timeSignatureFraction` gibt das neue Symbol an dieser Stelle aus.

Auf weitere Optionen kann man über die Scheme-Funktion `set-time-signature` zugreifen. In Kombination mit Abschnitt “Measure-grouping-engraver” in *Programmreferenz* werden hiermit Abschnitt “MeasureGrouping” in *Programmreferenz*-Zeichen erzeugt. Solche Zeichen erleichtern es, rhythmisch komplexe moderne Musik zu lesen. Im nächsten Beispiel ist der 9/8-Takt in 2, 2, 2 und 3 Achtel unterteilt. Das wird im dritten Argument an `set-time-signature` weitergegeben (2 2 2 3).

```
\score {
  \relative c'' {
    #(set-time-signature 9 8 '(2 2 2 3))
    g8[ g] d[ d] g[ g] a8[( bes g)] |
    #(set-time-signature 5 8 '(3 2))
    a4. g4
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```



## Siehe auch

Programmreferenz: Abschnitt “TimeSignature” in *Programmreferenz* und Abschnitt “Timing-translator” in *Programmreferenz*.

Beispiele: Abschnitt “Rhythms” in *Beispiele*.

## Bekannte Probleme und Warnungen

Automatische Balken richten sich nicht nach den Taktunterteilungen, die mit `set-time-signature` erzeugt werden.

### 1.2.3.2 Auftakte

Verkleinerte Takte, wie etwa ein Auftakt, werden wie folgt notiert:

```
\partial 16*5 c16 cis d dis e | a2. c,4 | b2
```



Die Syntax für den Befehl lautet:

`\partial` *Dauer*

wobei *Dauer* eine rhythmische Länge ist, die vor dem nächsten Taktstrich eingefügt wird.

Das wird intern übersetzt nach:

`\set Timing.measurePosition = -Länge der Dauer`

Die Eigenschaft `measurePosition` enthält eine rationale Zahl, die darstellt, wie groß der Abstand zum Taktanfang ist. Deshalb ist sie eine negative Zahl; `\partial 4` wird also intern übersetzt zu: „Eine Viertel bleibt übrig vom ganzen Takt.“

## Bekannte Probleme und Warnungen

Dieser Befehl berücksichtigt keine Verzierungen/Vorschläge am Anfang der Noten. Wenn ein Stück mit einem Vorschlag anfängt, muss der Befehl `\partial` *nach* dem Vorschlag kommen:

```
\grace f16
\partial 4
g4
a2 g2
```



`\partial` ist nur für den Anfang eines Stückes vorgesehen. Wenn der Befehl innerhalb eines Stückes verwendet wird, können seltsame Warnungen auftreten.

### 1.2.3.3 Musik ohne Metrum

Taktlinien und Taktzahlen werden automatisch erzeugt. Für Musik ohne Meter hingegen (etwa Kadenzen) ist das allerdings nicht erwünscht. Mit den Befehlen `\cadenzaOn` und `\cadenzaOff` kann dieses Verhalten ausgeschaltet und wieder angeschaltet werden.

```
c4 d e d
\cadenzaOn
c4 c d8 d d f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```



## Bekannte Probleme und Warnungen

LilyPond fügt Zeilen- und Seitenumbrüche nur an einer Taktlinie ein. Wenn die Kadenz nicht vor einem Umbruch endet, müssen Sie selber unsichtbare Taktlinien

`\bar ""`

einfügen, um anzuzeigen, wo umgebrochen werden darf.

### 1.2.3.4 Polymetrische Notation

### 1.2.3.5 Automatische Aufteilung von Noten

Lange Noten können automatisch in übergebundene Noten aufgeteilt werden. Dieses Verhalten erreicht man, indem der Abschnitt “*Note\_heads\_engraver*” in *Programmreferenz* mit dem Abschnitt “*Completion\_heads\_engraver*” in *Programmreferenz* ausgetauscht wird. Im nächsten Beispiel werden Noten, die über die Taktlinie dauern, aufgeteilt und übergebunden.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
} {
  c2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2
}
```



Dieser Notationsformatierer (eng. engraver) teilt alle Noten auf, die über eine Taktlinie dauern und fügt Bindebögen hinzu. Er kann unter Anderem dann nützlich sein, wenn man komplexe Partituren auf Fehler überprüfen möchte: Wenn die Takte nicht vollständig gefüllt sind, zeigt die Überbindung genau an, wie viele Notenwerte noch in dem jeweiligen Takt fehlen.

Wenn Sie wollen, dass auch Zeilenumbrüche an den Stellen, an denen automatisch Noten aufgeteilt wurden, stattfinden, müssen Sie auch den Formatierer Abschnitt “*Forbid\_line\_break\_engraver*” in *Programmreferenz* mit dem `\remove`-Befehl entfernen.

### Bekannte Probleme und Warnungen

Nicht alle Notenwerte (besonders wenn sie andere rhythmische Aufteilungen beinhalten) können exakt durch normale Noten und Punktierungen wiedergegeben werden. Der Engraver setzt aber trotzdem keine Triolen etc.

`Completion_heads_engraver` wirkt sich nur auf Noten aus; Pausen werden nicht aufgeteilt.

### Siehe auch

Programmreferenz: Abschnitt “*Completion\_heads\_engraver*” in *Programmreferenz*.

### 1.2.3.6 Melodierhythmus anzeigen

Manchmal soll nur der Rhythmus einer Melodie dargestellt werden. Das erreicht man mit einem Schlagzeug-Notensystem. Alle Tonhöhen werden auf eine Linie reduziert und das System hat auch nur eine einzige Linie.

```
\new RhythmicStaff {
  \time 4/4
  c4 e8 f g2 | r4 g r2 | g1:32 | r1 |
}
```



Siehe auch

Programmreferenz: Abschnitt “RhythmicStaff” in *Programmreferenz*.

### 1.2.4 Balken

### 1.2.4.1 Automatische Balken

LilyPond setzt Balken (engl. beam) automatisch.

$\backslash \text{time } 2/4 \text{ c8 c c c } \backslash \text{time } 6/8 \text{ c c c c8. c16 c8}$



Wenn diese automatischen Entscheidungen nicht gut genug sind, können die Balken auch explizit eingegeben werden. Es können auch bestimmte Balkenmuster, die sich vom Standard unterscheiden, definiert werden. Hierzu siehe den Abschnitt [Abschnitt 1.2.4.2 \[Einstellung von automatischen Balken\]](#), [Seite 22](#) für Einzelheiten.

Einzelne Noten können mit dem Befehl `\noBeam` markiert werden, damit sie nicht mit einem Balken versehen werden.

\time 2/4 c8 c\noBeam c c



Siehe auch

Programmreferenz: Abschnitt “Beam” in *Programmreferenz*.

#### 1.2.4.2 Einstellung von automatischen Balken

### 1.2.4.3 Manuelle Balken

In einigen Fällen kann es nötig sein, den automatischen Algorithmus für die Balken zu überschreiben. Die automatischen Balken werden beispielsweise nicht über Pausen oder Taktilinien hinweg gesetzt. Manuell definierte Balken werden mit den Zeichen [ und ] (AltGr+8 bzw. 9) markiert.

```
{
  r4 r8[ g' a r8] r8 g[ | a] r8
}
```





## Übliche Veränderungen der Einstellungen

LilyPond kann die Balkengruppen innerhalb eines größeren Balkens selber bestimmen, aber dabei können teilweise seltsame Ergebnisse entstehen. Mit den Eigenschaften `stemLeftBeamCount` und `stemRightBeamCount` können diese Unterteilungen manuell eingestellt werden. Wenn eine dieser Eigenschaften gesetzt ist, wird es nur einmal verwendet und dann wieder zum Standard zurück geschaltet.

```
{
  f8[ r16
    f g a]
  f8[ r16
    \set stemLeftBeamCount = #1
    f g a]
}
```



Die Eigenschaft `subdivideBeams` kann benutzt werden, um alle 16tel-Balken oder kleinere Werte zu bestimmten Taktzeiten zu unterteilen. Diese Zeiten werden mit der `beatLength`-Eigenschaft gesetzt.

```
c16[ c c c c c c c]
\set subdivideBeams = ##t
c16[ c c c c c c c]
\set Score.beatLength = #(ly:make-moment 1 8)
c16[ c c c c c c c]
```



Zu mehr Information über `make-moment` siehe [Abschnitt 1.2.6.3 \[Verwaltung der Zeiteinheiten\]](#), [Seite 29](#).

Zeilenumbrüche sind normalerweise verboten, wenn Balken sich über die Taktlinien erstrecken. Das kann aber durch Setzen von `breakable` verändert werden.

Balken mit Hälsen nach oben und unten werden automatisch eingesetzt, wenn ein großer Abstand zwischen Notenköpfen gefunden wird. Die Größe des Wertes kann mit dem `auto-knee-gap`-Objekt eingestellt werden.

## Bekannte Probleme und Warnungen

Automatisch erstellte Balken zwischen Systemen können nicht zusammen mit automatisch versteckten Systemen verwendet werden. Siehe auch [\[Systeme verstecken\]](#), [Seite 56](#).

Balken vermeiden nicht andere Objekte, wie etwa Text und Versetzungszeichen.

### 1.2.4.4 Gespreizte Balken

### 1.2.5 Takte

### 1.2.5.1 Taktlinien

Taktlinien trennen die Takte voneinander, werden aber auch verwendet, um Wiederholungen anzuzeigen. Normalerweise werden sie automatisch eingefügt. Zeilenumbrüche können nur an Taktlinien stattfinden.

Besondere Taktlinien-Arten können mit dem `\bar`-Befehl erzwungen werden.

```
c4 \bar "|:" c4
```



Folgende Taktlinienarten sind vorhanden:

Als letztes ist `"||:"` notiert, das sich ähnlich wie `"|:"` verhält. Es gibt jedoch nur an Zeilenenden eine doppelte Taktlinie aus und fängt die Wiederholungslinie erst in der nächsten Zeile an.

Um einen Zeilenumbruch an einer Stelle zu erlauben, wo keine sichtbare Taktlinie ist, kann man

```
\bar ""
```

benutzen. Damit wird eine unsichtbare Taktlinie an dieser Stelle eingefügt und damit ein Zeilenumbruch erlaubt (ohne dass sich die Anzahl der Takte erhöhen würde).

In Partituren mit vielen Systemen wird ein `\bar`-Befehl in einem System automatisch auf alle anderen Systeme angewendet. Die resultierenden Taktlinien sind miteinander verbunden innerhalb einer Gruppe (`StaffGroup`) oder einem Klaviersystem (`PianoStaff` bzw. (`GrandStaff`)).

```
<<
  \new StaffGroup <<
    \new Staff {
      e'4 d'
      \bar "||"
      f' e'
    }
    \new Staff { \clef bass c4 g e g }
  >>
  \new Staff { \clef bass c2 c2 }
>>
```



## Übliche Veränderungen der Einstellungen

Der Befehl `\bar Taktart` ist eine Kurzform von: `\set Timing.whichBar = Taktart`. Immer, wenn `whichBar` auf einen Wert gesetzt wird, wird eine Taktlinie dieses Typs erzeugt.

Eine Taktlinie wird auch durch Setzen der `whichBar`-Eigenschaft erzeugt. Am Anfang eines Taktes wird sie auf den Wert von `Timing.defaultBarType` gesetzt. Der Inhalt des `repeatCommands`-Befehls wird benutzt, um Standardtaktlinien zu überschreiben.

Sie sollten jedoch Wiederholungen mit dem `\repeat`-Befehl erzeugen. Siehe Abschnitt [Abschnitt 1.4 \[Wiederholungszeichen\]](#), Seite 39.

## Siehe auch

Im Handbuch: [Abschnitt 1.4 \[Wiederholungszeichen\]](#), Seite 39, [\[Klammern am Systemanfang\]](#), Seite 53.

Programmreferenz: [Abschnitt “BarLine” in Programmreferenz](#) (auf Partitur ([Abschnitt “Staff” in Programmreferenz](#))-Ebene erstellt), [Abschnitt “SpanBar” in Programmreferenz](#) (zwischen Systemen).

### 1.2.5.2 Taktzahlen

### 1.2.5.3 Takt- und Taktzahlüberprüfung

Die Taktüberprüfung hilft, Fehler in den Notendauern zu entdecken. Eine Taktüberprüfung wird mit dem Taktstrichsymbol „|“ (Taste `AltGr+<`) eingegeben. Immer, wenn LilyPond bei der Ausgabe des Notendrucks auf dieses Zeichen stößt, sollte hier in den Noten auch ein Taktstrich erscheinen. Wenn das nicht der Fall ist, wird eine Warnung ausgegeben. Im nächsten Beispiel resultiert die zweite Taktüberprüfung in einer Fehlermeldung.

```
\time 3/4 c2 e4 | g2 |
```

Taktüberprüfungen können auch in Liedtexten verwendet werden:

```
\lyricmode {
  \time 2/4
  Twin -- kle | Twin -- kle
}
```

Eine Taktüberprüfung gilt als nicht bestanden, wenn die Notenwerte nicht stimmen. Besonders in mehrstimmiger komplizierter Musik können solche falschen Notenwerte die ganze Partitur durcheinander bringen. Es lohnt sich also, die Fehlersuche damit zu beginnen, nicht bestandene Taktüberprüfungen zu kontrollieren.

Es ist auch möglich, die Bedeutung des Symbols `|` umzudefinieren. Das geschieht, indem man der Pipe (`pipeSymbol` einen musikalischen Ausdruck zuweist:

```
pipeSymbol = \bar "||"
```

```
{ c'2 c' | c'2 c' }
```



Wenn man größere Musikstücke kopiert, kann es hilfreich sein, wenn LilyPond überprüft, ob die Taktnummer, in der Sie gerade kopieren, mit der des Originalen übereinstimmt. Das kann mit dem Befehl `\barNumberCheck` folgenderweise überprüft werden:

`\barNumberCheck #123`

Eine Warnung wird ausgegeben, wenn der interne Zähler `currentBarNumber` von LilyPond nicht mit dem Wert 123 übereinstimmt.

#### 1.2.5.4 Übungszeichen

### 1.2.6 Besondere rhythmische Fragen

#### 1.2.6.1 Verzierungen

Verzierungen sind ausgeschriebene Ornamente. Die üblichste ist der Vorschlag. Er wird durch eine verkleinerte Note mit Schrägstrich und Bogen notiert. Der Vorhalt dagegen ist eine Verzierung, die einen bestimmten Notenwert der Hauptnote für sich beansprucht. Er wird als verkleinerte Note ohne Schrägstrich notiert. Der Vorschlag wird mit dem Befehl `\acciaccatura` und der Vorhalt mit `\appoggiatura` eingegeben.

```
b4 \acciaccatura d8 c4 \appoggiatura e8 d4
\acciaccatura { g16[ f] } e4
```



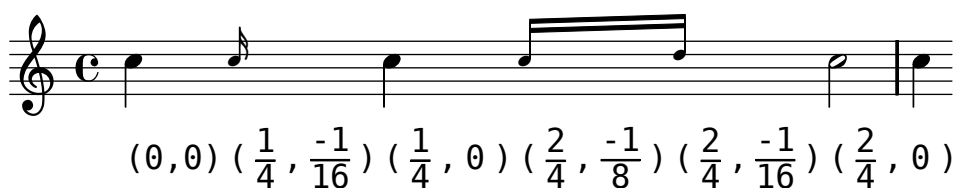
Bei beiden handelt es sich um spezielle Formen des `\grace` (engl. Verzierung)-Befehl. Wenn dieser Befehl einem musikalischen Ausdruck vorgestellt wird, wird ein neuer Ausdruck geschaffen, der in kleineren Noten gesetzt wird und von der logischen Zeit innerhalb des Taktes keinen Raum beansprucht.

```
c4 \grace c16 c4
\grace { c16[ d16] } c2 c4
```



Anders als `\acciaccatura` oder `\appoggiatura` setzt der `\grace`-Befehl keinen Bogen.

Programmintern wird die Zeitberechnung für Verzierungen in einer zweiten Zähllebene vorgenommen. Jeder Zeitpunkt beinhaltet zwei rationale Zahlen: die eine steht für die logische Zeit, die andere für die „Verzierungszeit“. Das obere Beispiel ist hier mit den entsprechenden Zeitwerten angezeigt:



Die Position von Verzierungen wird zwischen den Systemen synchronisiert. Im folgenden Beispiel sind jeweils zwei Sechzehntel gegen jede Achtel gesetzt:

```
<< \new Staff { e4 \grace { c16[ d e f] } e4 }
    \new Staff { c4 \grace { g8[ b] } c4 } >>
```



Eine Verzierung kann auch auf eine Note folgend gesetzt werden. Dazu wird der `\afterGrace`-Befehl benutzt. Er nimmt zwei Argumente: die Hauptnote und die Verzierungen, die nach der Hauptnote erscheinen sollen.

```
c1 \afterGrace d1 { c16[ d] } c4
```



Damit wird die Verzierung im Abstand von  $\frac{3}{4}$  der Länge der Hauptnote gesetzt. Dieser Bruch kann durch Setzen von `afterGraceFraction` verändert werden:

```
#(define afterGraceFraction (cons 7 8))
```

Hier wurde die Position auf das vorletzte Achtel der Notenlänge gesetzt.

Der gleiche Effekt kann auch manuell erreicht werden:

```
\new Voice {
  << { d1^\trill_( }
    { s2 \grace { c16[ d] } } >>
  c4)
}
```



Indem die Dauer der unsichtbaren Note (hier eine Halbe) wird der Abstand zwischen Hauptnote und Verzierung angepasst.

Ein `\grace`-Notenabschnitt wird nach besonderen Satzregeln gesetzt, um z. B. kleinere Noten zu benutzen und die Richtung der Hälse einzustellen. Veränderungen am Layout müssen also innerhalb des Verzierungsausdrucks gesetzt werden, damit sie auch eine Auswirkung haben.

```
\new Voice {
  \acciaccatura {
    \stemDown
    f16->
    \stemNeutral
  }
  g4
}
```



Diese Einstellungen müssen auch innerhalb der Verzierungsebene wieder rückgängig gemacht werden.

Das Layout der Verzierungsausdrücke kann mit der Funktion `add-grace-property` verändert werden. Im folgenden Beispiel wird die Richtung der Notenhälse neutralisiert, so dass sie nicht unbedingt in nach oben zeigen.

```
\new Staff {
  #(add-grace-property 'Voice 'Stem 'direction '())
  ...
}
```

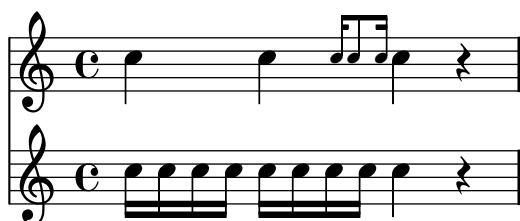
Eine andere Option ist es, die Variablen `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`, `stopAcciaccaturaMusic`, `startAppoggiaturaMusic` und `stopAppoggiaturaMusic` zu vermeiden. Mehr Information findet sich in der Datei `'ly/grace-init.ly'`.

Der Schrägstrich durch den Notenhals der Vorschläge kann auch in anderen Situation erreicht werden mit `\override Stem #'stroke-style = #"grace"`.

## Übliche Veränderungen der Einstellungen

Verzierungen könne gezwungen werden, den Hauptnoten entsprechend aufgeteilt zu werden.

```
<<
\override Score.SpacingSpanner #'strict-grace-spacing = ##t
\new Staff {
  c4
  \afterGrace c4 { c16[ c8 c16] }
  c4 r
}
\new Staff {
  c16 c c c c c c c c4 r
}
>>
```



## Siehe auch

Programmreferenz: [Abschnitt “GraceMusic” in Programmreferenz](#).

## Bekannte Probleme und Warnungen

Eine Partitur, die mit einem `\grace`-Ausdruck beginnt, benötigt eine explizit gesetzte neue Stimme (`\new Voice`), sonst werden Hauptnote und Verzierung auf verschiedenen Systemen gesetzt.

Die Synchronisation von Verzierungen kann auch zu Überraschungen führen. Auch andere Symbole der Systeme, wie Vorzeichen, Taktlinien usw., werden synchronisiert. Vorsicht ist geboten, wenn nur in bestimmten Systemen Verzierungen vorkommen:

```
<< \new Staff { e4 \bar "|:" \grace c16 d4 }
    \new Staff { c4 \bar "|:" d4 } >>
```



Dem kann abgeholfen werden, indem unsichtbare Verzierungsnoten der selben Länge in die anderen Systeme gesetzt werden. Im obigen Beispiel müsste also

```
<< \new Staff { e4 \bar "|" \grace c16 d4 }
    \new Staff { c4 \bar "|" \grace s16 d4 } >>
```



gesetzt werden.

Verzierungsabschnitte sollten nur innerhalb von sequentiellen musikalischen Ausdrücken benutzt werden. Wenn sie ineinandergeschachtelt werden, kann es zu Fehlermeldungen oder Abstürzen kommen.

### 1.2.6.2 An Kadenzen ausrichten

### 1.2.6.3 Verwaltung der Zeiteinheiten

## 1.3 Ausdrucksbezeichnungen

RONDO  
*Allegro*

### 1.3.1 An Noten angehängt

#### Artikulationszeichen und Verzierungen

Eine Vielfalt an Symbolen kann über und unter den Noten erscheinen, um zu markieren, auf welche Art die Note ausgeführt werden soll. Sie werden in LilyPond notiert, indem ein Minuszeichen an die Note gehängt wird, gefolgt von dem jeweiligen Zeichen. Hier einige Beispiele:

Die Bedeutung der Zeichen kann auch verändert werden. Siehe etwa `'ly/script-init.ly'` für Beispiele.

Das Artikulationszeichen wird automatisch gesetzt, aber die Richtung kann auch erzwungen werden. Wie auch bei anderen LilyPond-Befehlen, erreicht man mit `_` eine Ausrichtung unter der Note, mit `^` eine Ausrichtung über der Note.

```
c''4^^ c''4_~
```

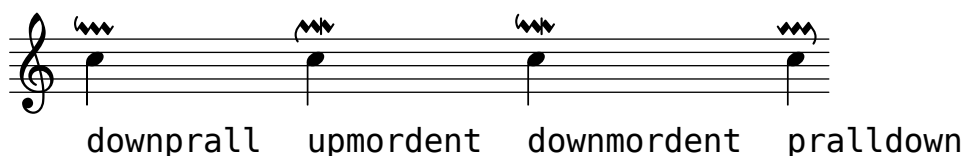
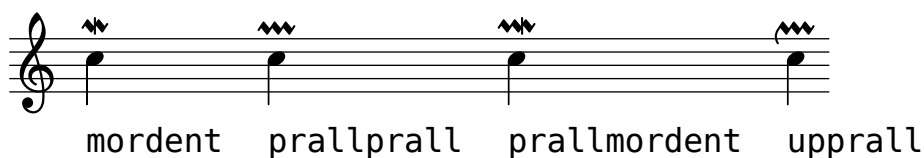
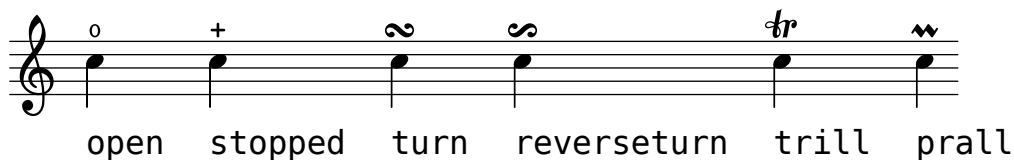
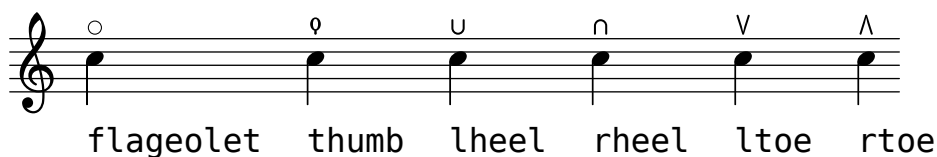
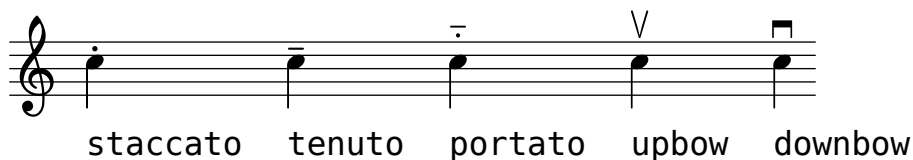
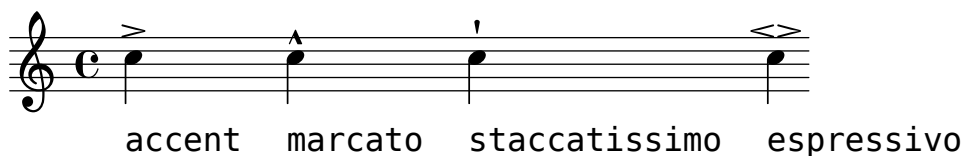


Andere Symbole können mit der Syntax `Note\Bezeichnung` hinzugefügt werden. Auch sie können mit `^` und `_` nach oben und unten gezwungen werden:

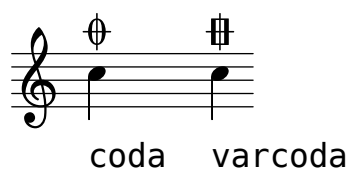
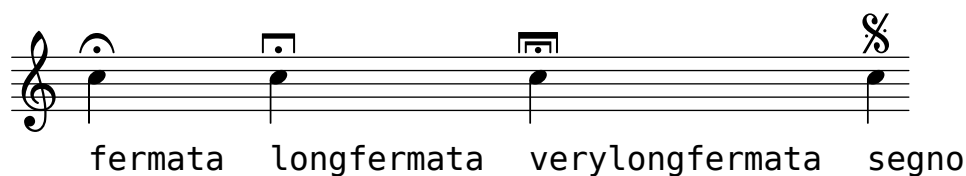
```
c\fermata c^\fermata c_\'fermata
```



Hier ist eine Liste, die alle möglichen Zeichen darstellt:







## Übliche Veränderungen der Einstellungen

Die vertikale Anordnung der Zeichen wird durch die `script-priority`-Eigenschaft kontrolliert. Je kleiner die Zahl ist, umso näher wird das Zeichen an die Note gesetzt. In dem nächsten Beispiel hat das Textsymbol ([Abschnitt “TextScript” in Programmreferenz](#)), ein Kreuz, die niedrigste Priorität und wird also als unterstes gesetzt. Im zweiten Beispiel hat der Praller (das [Abschnitt “Script” in Programmreferenz](#)) die niedrigste Priorität und erscheint innen. Wenn zwei Objekte die gleiche Priorität haben, entscheidet die Reihenfolge, in der sie notiert sind, welches zuerst kommt.

```
\once \override TextScript #'script-priority = #-100
a4^\prall^\markup { \sharp }
```

```
\once \override Script #'script-priority = #-100
a4^\prall^\markup { \sharp }
```



## Siehe auch

Programmreferenz: [Abschnitt “Script” in Programmreferenz](#).

## Bekannte Probleme und Warnungen

Diese Zeichen erscheinen zwar im Druck, haben aber keine Auswirkung auf die produzierte MIDI-Datei.

## Dynamik

Absolute Dynamikbezeichnung wird mit Befehlen nach den Noten angezeigt. Die vordefinierten Befehle lauten: `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz`, and `\rfz`.

```
c\ppp c\pp c \p c\mp c\mf c\f c\ff c\fff
c2\fp c\sf c\sff c\sp c\spp c\sfz c\rfz
```



Eine Crescendo-Klammer wird mit dem Befehl `\<` begonnen und mit `\!` oder einem absoluten Dynamikbefehl beendet. Ein Decrescendo beginnt mit `\>` und wird auf die gleiche Art beendet. `\cr` und `\decr` können anstelle von `\<` und `\>` benutzt werden. Weil diese Zeichen an Noten gekoppelt sind, müssen unsichtbare Noten benutzt werden, wenn mehr als ein Zeichen pro Note benötigt wird.

```
c\< c\! d\> e\!
<< f1 { s4 s4\< s4\! \> s4\! } >>
```



Eine Crescendo-Klammer beginnt normalerweise am linken Rand der Anfangsnote und endet am rechten Rand der Endnote. Wenn das Ende auf den Taktanfang fällt, endet die Klammer an der direkt vorhergehenden Taktlinie. Diese Einstellung lässt sich aber durch die Eigenschaft `hairpinToBarline` verändern.

In manchen Situationen kann auch der `\espressivo`-Befehl geeignet sein, ein An- und Abschwellen einer Note anzuzeigen.

```
c2 b4 a g1\espressivo
```



Hier können allerdings sehr kurze Klammern auftreten. Der Wert von `minimum-length` in [Abschnitt "Voice" in Programmreferenz](#). Abschnitt "Hairpin" in [Programmreferenz](#) kann angepasst werden, etwa:

```
\override Voice.Hairpin #'minimum-length = #5
```

Klammern können auch mit einem kleinen Kreis um die Spitze (al niente-Notation) gedruckt werden, wenn die `circled-tip`-Eigenschaft gesetzt wird.

```
\override Hairpin #'circled-tip = ##t
c2\< c\!
c4\> c\< c2\!
```



Anstelle der Klammern kann auch der Text *cresc.* bzw. *decr.* oder *dim.* ausgegeben werden.

```
\crescTextCresc
c\< d e f\!
\crescHairpin
e\> d c b\!
\dimTextDecresc
c\> d e f\!
\dimTextDim
e\> d c b\!
```



Dieser Text kann auch beliebig angepasst werden:

```
\set crescendoText = \markup { \italic "cresc. poco" }
\set crescendoSpanner = #'text
a'2\< a a a\!\mf
```



Um neue Dynamikzeichen oder Text, der mit diesen zusammen gesetzt wird, zu erstellen, siehe den Abschnitt [\[Neue Lautstärkezeichen\]](#), Seite 34.

Vertikale Position der Zeichen wird von der Funktion [Abschnitt “DynamicLineSpanner”](#) in [Programmreferenz](#) verwaltet.

## Übliche Veränderungen der Einstellungen

Dynamikzeichen, die an der selben Note auftreten, werden vertikal angeordnet. Wenn Sie sicher gehen wollen, dass die Zeichen angeordnet werden, auch wenn sie nicht an der selben Note vorkommen, kann die `staff-padding`-Eigenschaft vergrößert werden.

```
\override DynamicLineSpanner #'staff-padding = #4
```

Diese Eigenschaft kann man auch benutzen, um Dynamikzeichen davor zu hindern, mit anderen Noten zusammenzustößen.

Crescendi and Decrescendi, die an der ersten Note einer neuen Zeile enden, werden nicht ausgegeben. Mit

```
\override Score.Hairpin #'after-line-breaking = ##t
```

wird dieses Verhalten ausgeschaltet.

Text für dynamische Änderungen (wie *cresc.*) wird mit einer gestrichelten Linie gesetzt. Um diese Linie zu unterdrücken, kann der Befehl

```
\override DynamicTextSpanner #'dash-period = #-1.0
```

eingesetzt werden.

## Vordefinierte Befehle

`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`.

## Siehe auch

Programmreferenz: Abschnitt “DynamicText” in *Programmreferenz*, Abschnitt “Hairpin” in *Programmreferenz*. Vertikale Positionierung der Symbole wird von der Eigenschaft Abschnitt “DynamicLineSpanner” in *Programmreferenz* verwaltet.

## Neue Lautstärkezeichen

### 1.3.2 Bögen

#### Legatobögen

Ein Legatobogen (engl. slur) zeigt an, dass die Noten *legato* gespielt werden sollen. Er wird mit Klammern hinter den Notenwerten notiert.

```
f( g a) a8 b( a4 g2 f4)
<c e>2( <b d>2)
```



Die Richtung eines Legatobogens kann mit den Befehlen `\slurDIR`, wobei *DIR* entweder Up, Down, oder Neutral, angezeigt werden.

Es gibt aber auch eine Kurzform. Indem `_` oder `^` for die öffnende Klammer gestellt wird, wird die Richtung angegeben.

```
c4_( c) c^( c)
```



Nur ein Legatobogen kann gleichzeitig geschrieben werden. Wenn Sie einen langen Bogen über mehreren kurzen notieren wollen, müssen Sie [\[Phrasierungsbögen\]](#), Seite 35 benutzen.

## Übliche Veränderungen der Einstellungen

Manche Komponisten schreiben zwei Legatobögen, um Legatoakkorde zu markieren. Das kann in LilyPond erreicht werden, indem die Eigenschaft `doubleSlurs` gesetzt wird.

```
\set doubleSlurs = ##t
<c e>4 ( <d f> <c e> <d f> )
```



## Vordefinierte Befehle

`\slurUp`, `\slurDown`, `\slurNeutral`, `\slurDashed`, `\slurDotted`, `\slurSolid`.

## Siehe auch

Programmreferenz: [Abschnitt “Slur” in Programmreferenz](#).

## Phrasierungsbögen

Ein Phrasierungsbogen verbindet Noten und wird verwendet, um einen musikalischen Ausdruck anzuzeigen. Er wird mit den Befehlen `\(` und `\)` eingegeben.

`\time 6/4 c' \( d( e) f( e) d\)`



Im typographischen Sinne verhalten sich Phrasierungsbögen genauso wie Legatobögen. Sie werden aber als eigene Objekte behandelt. Ein `\slurUp` hat also keine Auswirkung auf die Phrasierungsbögen, anstelle dessen muss `\phrasingSlurUp`, `\phrasingSlurDown` oder `\phrasingSlurNeutral` benutzt werden.

Es können keine simultanen Phrasierungsbögen gesetzt werden.

## Vordefinierte Befehle

`\phrasingSlurUp`, `\phrasingSlurDown`, `\phrasingSlurNeutral`.

## Siehe auch

Programmreferenz: [Abschnitt “PhrasingSlur” in Programmreferenz](#).

## Laissez-vibrer-Bögen

So genannte „laissez vibrer“-Bögen werden verwendet um anzuzeigen, dass man die Musik ausklingen lassen soll. Sie werden in der Klavier-, Harfen-, anderer Saiteninstrument- und Schlagzeugnotation verwendet. Sie können mit dem Befehl `\laissezVibrer` eingegeben werden.

`<c f g>\laissezVibrer`



## Siehe auch

Programmreferenz: [Abschnitt “LaissezVibrerTie” in Programmreferenz](#), [Abschnitt “LaissezVibrerTieColumn” in Programmreferenz](#).

Beispiele: [Abschnitt “Expressive marks” in Beispiele](#).

## Atemzeichen

Atemzeichen werden mit dem Befehl `\breathe` eingegeben.

```
c'4 \breathe d4
```



## Übliche Veränderungen der Einstellungen

Das Symbol für das Atemzeichen kann verändert werden, indem die Eigenschaft `text` des `BreathingSign`-Objektes mit beliebigem Text überschrieben wird. Zum Beispiel ergibt

```
c'4
\override BreathingSign #'text
  = #(make-musicglyph-markup "scripts.rvarcomma")
\breathe
d4
```



## Siehe auch

Programmreferenz: [Abschnitt “BreathingSign” in Programmreferenz.](#)

Beispiele: [Abschnitt “Winds” in Beispiele.](#)

## Glissando zu unbestimmter Tonhöhe

Gleiten nach oben und unten kann mit dem Befehl `\bendAfter` notiert werden.



### 1.3.3 Linien

#### Glissando

Ein Glissando ist ein Gleiten zwischen Tonhöhen. Es wird mit einer geraden oder gezackten Linie zwischen zwei Noten notiert. Es wird mit dem Befehl `\glissando` auf eine Note folgend notiert.

```
c2\glissando c'
\override Glissando #'style = #'zigzag
c2\glissando c,
```



## Übliche Veränderungen der Einstellungen

```
I = \once \override NoteColumn #'ignore-collision = ##t
```

```
\relative <<
  { \oneVoice \stemDown f2 \glissando \stemNeutral a } \\  
  { \oneVoice \I c2 \glissando \I d, }
>>
```



## Siehe auch

Programmreferenz: [Abschnitt “Glissando” in Programmreferenz.](#)

Beispiele: [Abschnitt “Expressive marks” in Beispiele.](#)

## Bekannte Probleme und Warnungen

Text über der Linie (wie etwa *gliss.*) wird nicht unterstützt.

## Arpeggio

Ein Arpeggio als Zeichen, dass ein Akkord gebrochen gespielt werden soll, kann mit dem Befehl `\arpeggio` hinter dem Akkord erzeugt werden.

```
<c e g c>\arpeggio
```



Eine eckige Klammer zur Linken des Akkordes zeigt an, dass kein Arpeggio gespielt werden soll.

```
\arpeggioBracket  
<c' e g c>\arpeggio
```



Die Richtung des Arpeggios wird manchmal mit Pfeilen notiert und hat eigene Befehle.

```
\new Voice {  
  \arpeggioArrowUp  
  <c e g c>\arpeggio  
  \arpeggioArrowDown  
  <c e g c>\arpeggio  
}
```



## Übliche Veränderungen der Einstellungen

Wenn ein Arpeggio sich über mehrere Systeme erstreckt, kann mit einem Klaviersystem die Eigenschaft **Abschnitt “PianoStaff” in Programmreferenz**.`connectArpeggios` gesetzt werden.

```
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff { <c' e g c>\arpeggio }
  \new Staff { \clef bass <c,, e g>\arpeggio }
>>
```



## Vordefinierte Befehle

`\arpeggio`, `\arpeggioArrowUp`, `\arpeggioArrowDown`, `\arpeggioNormal`, `\arpeggioBracket`.

## Siehe auch

Notationshandbuch: **Abschnitt 1.2.1.4 [Bindebögen]**, Seite 15, um Arpeggios auszuschreiben.

Programmreferenz: **Abschnitt “Arpeggio” in Programmreferenz**.

## Bekannte Probleme und Warnungen

Es ist nicht möglich, Arpeggios zwischen Systemen und solche, die sich nur auf ein System erstrecken, zum gleichen Zeitpunkt in einem Klaviersystem zu benutzen.

## Triller

Kurze Triller können wie ein normales Artikulationszeichen eingegeben werden, siehe **[Artikulationszeichen und Verzierungen]**, Seite 29.

Längere Triller werden mit den Befehlen `\startTrillSpan` zu Beginn und `\stopTrillSpan` am Ende erstellt.

```
\new Voice {
  << { c1 \startTrillSpan }
      { s2. \grace { d16[\stopTrillSpan e] } } } >>
  c4 }
```





Triller, die auf einer bestimmten Note ausgeführt werden sollen, können mit dem Befehl `pitchedTrill` notiert werden.

```
\pitchedTrill c4\startTrillSpan fis
f\stopTrillSpan
```



Dabei ist das erste Argument die Hauptnote. Die zweite Note wird ohne Hals in Klammern gesetzt.

## Vordefinierte Befehle

```
\startTrillSpan, \stopTrillSpan.
```

## Siehe auch

Programmreferenz: [Abschnitt “TrillSpanner” in Programmreferenz.](#)

## 1.4 Wiederholungszeichen



Wiederholung ist ein zentrales Konzept in der Musik, und es gibt eine ganze Vielzahl von Notationsmöglichkeiten für Wiederholungen.

### 1.4.1 Lange Wiederholungen eingeben

#### Wiederholungstypen

Die folgenden Wiederholungsarten sind unterstützt:

**unfold** Die wiederholte Musik wird vollständig ausgeschrieben (bzw. gespielt). Hiermit können sehr einfach sich wiederholende Stellen notiert werden. Es ist auch der einzige Wiederholungstyp, der in der MIDI-Ausgabe berücksichtigt wird.

<b>volta</b>	Wiederholungen werden nicht ausgeschrieben, aber alternative Endungen (Volta-Klammern) können bei Bedarf notiert werden. Das ist die übliche Wiederholung für Wiederholungen mit unterschiedlichen Enden. Die Wiederholung wird in der MIDI-Datei nicht berücksichtigt.
<b>tremolo</b>	Hiermit können Tremolo-Balken erstellt werden. Sie werden nicht in die MIDI-Datei aufgenommen.
<b>percent</b>	Hiermit können noten- oder taktweise Wiederholungszeichen notiert werden. Sie erinnern an das Prozentzeichen. Auch sie werden nicht in der MIDI-Datei berücksichtigt. Diese Wiederholungen müssen innerhalb eines Stimmen (Voice)-Kontextes erstellt werden.

## Die Syntax der Wiederholungen

LilyPond besitzt eine einzige Syntax für alle unterschiedlichen Wiederholungstypen. Sie lautet:

```
\repeat Typ Wiederholungszähler Wiederholungsnoten
```

Wenn Sie unterschiedliche Endungen haben, können Sie diese mit dem Befehl

```
\alternative {
  Klammer1
  Klammer2
  Klammer3
  ...
}
```

wobei jede Klammer ein musikalischer Ausdruck ist. Wenn Sie nicht genug unterschiedliche Alternativen für alle Wiederholungen angeben, wird angenommen, dass die erste Alternative mehr als einmal verwendet wird.

Eine typische Wiederholung kann so aussehen:

```
c1
\repeat volta 2 { c4 d e f }
\repeat volta 2 { f e d c }
```



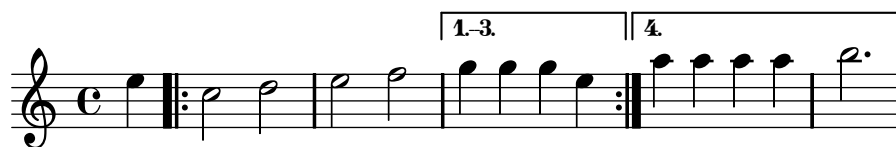
Und mit unterschiedlichen Klammern:

```
c1
\repeat volta 2 {c4 d e f}
\alternative { {d2 d} {f f,} }
```



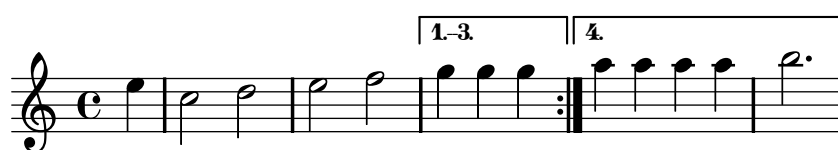
Wiederholungen können mit Auftakten kombiniert werden.

```
\new Staff {
  \partial 4 e |
  \repeat volta 4 { c2 d2 | e2 f2 | }
  \alternative { { g4 g g e } { a a a a | b2. } }
}
```



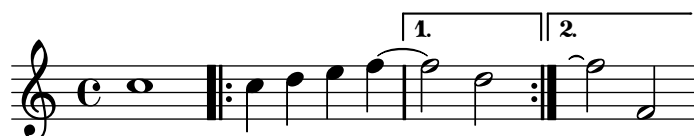
or

```
\new Staff {
  \partial 4
  \repeat volta 4 { e | c2 d2 | e2 f2 | }
  \alternative { { \partial 4*3 g4 g g } { a a a a | b2. } }
}
```



Bindebögen können auch an eine zweite Klammer angefügt werden.

```
c1
\repeat volta 2 { c4 d e f ~ }
\alternative { { f2 d } { f\repeatTie f, } }
```



Es ist auch möglich, die Klammern zu verkürzen, indem ihnen ein Wert in der Eigenschaft `voltaSpannerDuration` zugewiesen wird. Im nächsten Beispiel ist die Klammer beispielsweise nur einen 3/4-Takt lang.

```
\relative c''{
  \time 3/4
  c c c
  \set Score.voltaSpannerDuration = #(ly:make-moment 3 4)
  \repeat volta 5 { d d d }
  \alternative { { e e e f f f }
    { g g g } }
}
```



Wenn eine Wiederholung am Anfang einer neuen Zeile beginnen soll und eine Doppellinie am Ende der vorhergehenden Zeile stehen soll, muss

```
... \bar "||:" \break
\repeat volta 2 { ...
```

benutzt werden. Siehe [Abschnitt 1.2.5.1 \[Taktlinien\]](#), [Seite 24](#) for more information.

## Siehe auch

Programmreferenz: Abschnitt “VoltaBracket” in *Programmreferenz*, Abschnitt “Repeated-Music” in *Programmreferenz*, Abschnitt “VoltaRepeatedMusic” in *Programmreferenz* und Abschnitt “UnfoldedRepeatedMusic” in *Programmreferenz*.

Beispiele:

Klammern für die Wiederholung werden normalerweise nur über dem obersten System ausgegeben. Das kann verändert werden, indem `Volta_engraver` in den `Staff`-Kontext gesetzt wird, über dem diese Klammern erscheinen sollen. Vgl. Abschnitt 5.1.3 [Umgebungs-Plugins verändern], Seite 110 und Abschnitt “Repeats” in *Beispiele*.

## Bekannte Probleme und Warnungen

Eine ineinandergeschachtelte Wiederholung wie

```
\repeat ...
\repeat ...
\alternative
```

ist mehrdeutig, weil nicht klar ist, zu welchem `\repeat`-Abschnitt die `\alternative`-Endung gehört. Diese Mehrdeutigkeit wird von LilyPond aufgelöst, indem die alternative Endung immer zu der innersten Wiederholung gehört. Um Klarheit zu schaffen, bietet es sich an, in solchen Situationen Klammern zu benutzen.

Die Taktposition wird bei einer alternativen Endung nicht mitgeteilt, so dass nach einer Wiederholung diese Information manuell angegeben werden muss, entweder durch setzen von `Score.measurePosition` oder indem der Befehl `\partial` benutzt wird. Gleichmaßen werden auch Legato- oder Bindebögen nicht wiederholt.

## Manuelle Wiederholungsbefehle

Die Eigenschaft `repeatCommands` kann verwendet werden, um das Aussehen der Wiederholungen zu beeinflussen. Ihr Argument ist eine Scheme-Liste an Wiederholungsbefehlen.

`start-repeat`

Setzt eine |: Taktlinie.

`end-repeat`

Setzt eine :| Taktlinie.

`(volta text)`

Setzt eine Volta-Klammer mit der Beschriftung *text*: Der Text kann definiert werden als Textstring oder formatierter Text, siehe Abschnitt [Abschnitt 1.8.2 \[Text formatieren\]](#), Seite 58. Es darf nicht vergessen werden, die Schriftart zu verändern, weil die Standardschriftart für die Nummern keine Buchstaben enthält.

`(volta #f)`

Beendet eine aktive Klammer.

`c4`

```
\set Score.repeatCommands = #'((volta "93") end-repeat)
```

`c4 c4`

```
\set Score.repeatCommands = #'((volta #f))
```

`c4 c4`



## Siehe auch

Programmreferenz: Abschnitt “VoltaBracket” in *Programmreferenz*, Abschnitt “Repeated-Music” in *Programmreferenz*, Abschnitt “VoltaRepeatedMusic” in *Programmreferenz* und Abschnitt “UnfoldedRepeatedMusic” in *Programmreferenz*.

### 1.4.2 Kurze Wiederholungen

#### Prozent-Wiederholungen

Wenn der Prozent (`percent`)-Wiederholungsstil gesetzt ist, wird eine Notenphrase wiederholt. Sie wird einmal gedruckt und dann durch ein spezielles Zeichen ersetzt. Phrasen von ein oder zwei Takten Dauer werden durch ein dem Prozentzeichen ähnlichen Zeichen markiert, Phrasen, die kürzer als ein Takt sind, durch einen Schrägstrich dargestellt. Dieser Wiederholungstyp muss innerhalb eines Stimmen (Voice)-Kontextes notiert werden.

```
\new Voice \relative c' {
  \repeat percent 4 { c4 }
  \repeat percent 2 { c2 es2 f4 fis4 g4 c4 }
}
```



Wiederholungen, die länger als einen Takt dauern, können gezählt werden, wenn die `countPercentRepeats`-Eigenschaft eingeschaltet wird.

```
\new Voice {
  \set countPercentRepeats = ##t
  \repeat "percent" 4 { c1 }
}
```



Isolierte Prozentzeichen können auch gedruckt werden. Das geschieht, indem einer Ganztakt-pause (R) eine andere Funktion zugewiesen wird.

```
\override MultiMeasureRest #'stencil
= #ly:multi-measure-rest::percent
R1
```



## Siehe auch

Programmreferenz: Abschnitt “RepeatSlash” in *Programmreferenz*, Abschnitt “PercentRepeat” in *Programmreferenz*, Abschnitt “DoublePercentRepeat” in *Programmreferenz*, Abschnitt “DoublePercentRepeatCounter” in *Programmreferenz*, Abschnitt “PercentRepeatCounter” in *Programmreferenz*, Abschnitt “PercentRepeatedMusic” in *Programmreferenz*.

## Tremolo-Wiederholung

Um Tremolozeichen zwischen den Noten zu setzen, kann der Wiederholungstyp `tremolo` benutzt werden.

```
\new Voice \relative c' {
  \repeat tremolo 8 { c16 d16 }
  \repeat tremolo 4 { c16 d16 }
  \repeat tremolo 2 { c16 d16 }
}
```



Tremolozeichen können auch einer einzelnen Noten hinzugefügt werden. In diesem Fall darf die Note nicht von Klammern eingefasst sein.

```
\repeat tremolo 4 c'16
```



Ähnliche Darstellung wird erreicht durch eine innere Tremolounterteilung, die im Abschnitt [\[Tremolo-Unterteilung\]](#), Seite 44 beschrieben wird.

## Siehe auch

Im Handbuch: [\[Tremolo-Unterteilung\]](#), Seite 44, Abschnitt 1.4 [\[Wiederholungszeichen\]](#), Seite 39.

Programmreferenz: Abschnitt “Beam” in *Programmreferenz*, Abschnitt “StemTremolo” in *Programmreferenz*.

## Tremolo-Unterteilung

Tremolozeichen können einer einzelnen Noten hinzugefügt werden, indem an sie die Zeichen `: [Anzahl]` angefügt werden. Die Anzahl bezeichnet die Dauer der einzelnen Noten, und ihr Mindestwert ist 8. Mit der Zahl 8 erhält man eine Linie durch den Notenhals. Wenn die Anzahl ausgelassen wird, wird der letzte benutzte Wert (in der Funktion `tremoloFlags` gespeichert) eingesetzt.

```
c'2:8 c':32 | c': c': |
```



## Bekannte Probleme und Warnungen

Tremolos, die auf diese Weise notiert werden, werden nicht in die MIDI-Datei aufgenommen.

## Siehe auch

Im Handbuch: [\[Tremolo-Wiederholung\]](#), Seite 44.

Programmreferenz: Abschnitt “StemTremolo” in *Programmreferenz*.

## 1.5 Gleichzeitig erscheinende Noten



Polyphonie bedeutet in der musikalischen Terminologie das Vorhandensein von mehr als einer (eigenständigen) Stimme in einem Stück. Für LilyPond bedeutet es aber das Vorhandensein von mehr als einer Stimme pro System.

### 1.5.1 Eine einzelne Stimme

#### Noten mit Akkorden

Ein Akkord wird notiert, indem die zu ihm gehörenden Tonhöhen zwischen spitze Klammern (< und > gesetzt werden. Auf einen Akkord kann eine Dauer-Angabe folgen, genauso wie bei einfachen Noten.

<c e g>4 <c>8



Siehe [Abschnitt 2.7 \[Notation von Akkorden\]](#), [Seite 77](#) für mehr Information.

#### Cluster

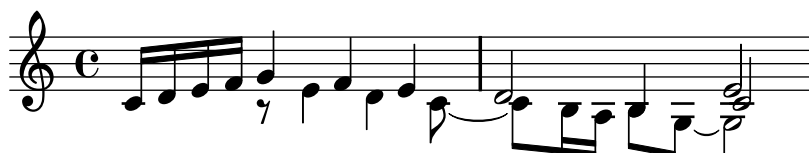
### 1.5.2 Mehrere Stimmen

#### Einfache Mehrstimmigkeit

Die einfachste Weise, Abschnitte mit mehr als einer Stimme pro Notensystem zu notieren, ist es, jede Stimme als eine Sequenz zu notieren (innerhalb der Klammern {...}) und dann die beiden Klammern simultan zu kombinieren, indem sie mit \\ getrennt werden.

```
\new Staff \relative c' {
  c16 d e f
  <<
  { g4 f e | d2 e2 } \\
```

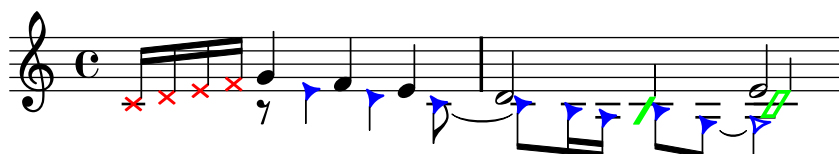
```
{ r8 e4 d c8 ~ | c b16 a b8 g ~ g2 } \\
{ s2. | s4 b4 c2 }
>>
}
```



Dieser Trenner veranlasst, dass Stimmen (**Abschnitt “Voice” in *Programmreferenz***) -Kontexte<sup>2</sup> angelegt werden. Sie tragen die Namen "1", "2" usw. In jedem dieser Kontexte wird die Richtung von Bögen, Hälsen usw. entsprechend angepasst.

Diese Stimmen sind alle unabhängig von der Stimme, in der die Noten außerhalb der << \\ >>-Konstruktion notiert sind. Das sollte man berücksichtigen, wenn man auf Stimmen-Ebene Veränderungen vornimmt. Das bedeutet gleichzeitig auch, dass Legato- und Bindebögen nicht in eine << \\ >>-Umgebung hinein- noch aus hier hinaus ragen können. Die parallelen Notenabschnitte aus unterschiedlichen << \\ >>-Umgebungen hingegen gehören der gleichen Stimme an. Hier noch einmal das gleiche Beispiel mit unterschiedlichen Notenköpfen und Farben für jede Stimme. Die Veränderung der Notenköpfe in der Hauptstimme hat keine Auswirkung auf die Stimmen innerhalb der << \\ >>-Umgebungen, und die Veränderung der Notenköpfe für die untere Stimme setzt sich fort in der zweiten << \\ >>-Umgebung. Hier ist auch eine Note über die Taktgrenze hinweg an die untere Stimme der zweiten Polyphonie-Umgebung angebunden.

```
\new Staff \relative c' {
  \override NoteHead #'style = #'cross
  \override NoteHead #'color = #red
  c16 d e f
  <<
    { g4 f e } \\
    { \override NoteHead #'style = #'triangle
      \override NoteHead #'color = #blue
      r8 e4 d c8 ~ }
  >> |
  <<
    { d2 e2 } \\
    { c8 b16 a b8 g ~ g2 } \\
    { \override NoteHead #'style = #'slash
      \override NoteHead #'color = #green
      s4 b4 c2 }
  >>
}
```



Polyphonie verändert nicht das Verhältnis der Noten innerhalb einer `\relative { }`-Umgebung. Jede Note wird weiterhin errechnet aus der direkt vorhergehenden.

<sup>2</sup> Polyphone Stimmen werden in anderen Programmen teilweise als „layers“ (Schichten) bezeichnet.



```
\relative { NoteA << NoteB \\ NoteC >> NoteD }
```

NoteC ist relativ zu NoteB, nicht NoteA; NoteD ist relativ zu NoteC, nicht NoteB oder NoteA.

## Stimmen explizit beginnen

**Abschnitt “Voice”** in *Programmreferenz*-Kontexte können auch manuell innerhalb eines << >>-Abschnittes initiiert werden. Mit den Befehlen `\voiceOne` bis hin zu `\voiceFour` kann jeder Stimme entsprechendes Verhalten von vertikaler Verschiebung und Richtung von Hälsen und anderen Objekten hinzugefügt werden.

Genauer gesagt,

```
<< \upper \\ \lower >>
```

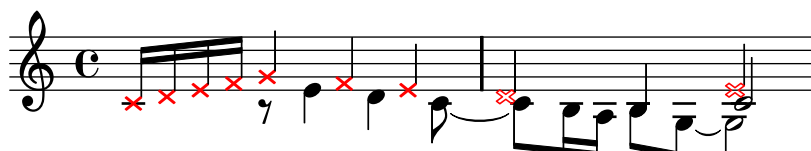
entspricht

```
<<
  \new Voice = "1" { \voiceOne \upper }
  \new Voice = "2" { \voiceTwo \lower }
>>
```

Der `\voiceXXX`-Befehl setzt die Richtung von Hälsen, Bögen, Artikulationszeichen, Text, Punktierungen und Fingersätzen. `\voiceOne` und `\voiceThree` lassen diese Objekte nach oben zeigen, `\voiceTwo` und `\voiceFour` dagegen lassen sie abwärts zeigen. Der Befehl `\oneVoice` stellt wieder auf das normale Verhalten um.

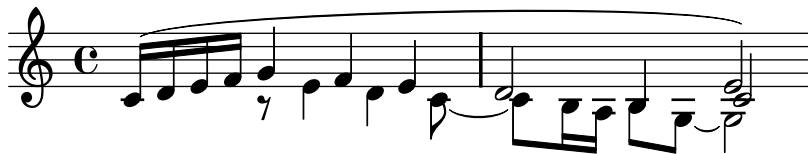
Ein Ausdruck, der direkt innerhalb einer << >>-Umgebung auftritt, gehört der Hauptstimme an. Das ist nützlich, wenn zusätzliche Stimme auftreten, während die Hauptstimme sich fortsetzt. Hier also eine bessere Version des Beispiels aus dem vorigen Abschnitt. Die farbigen Kreuz-Notenköpfe zeigen, dass die Hauptstimme sich jetzt in einem einzigen Stimmen (voice)-Kontext befindet.

```
\new Staff \relative c' {
  \override NoteHead #'style = #'cross
  \override NoteHead #'color = #red
  c16 d e f
  \voiceOne
  <<
    { g4 f e | d2 e2 }
    \new Voice="1" { \voiceTwo
      r8 e4 d c8 ~ | c8 b16 a b8 g ~ g2
      \oneVoice
    }
    \new Voice { \voiceThree
      s2. | s4 b4 c2
      \oneVoice
    }
  >>
  \oneVoice
}
```



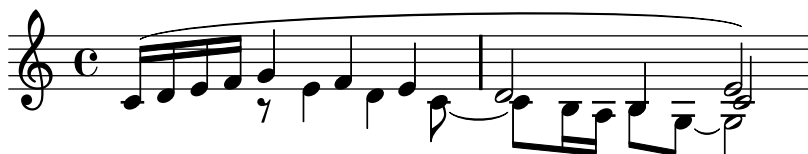
Und mit der richtigen Definition der Stimmen kann die Melodie auch übergeben werden.

```
\new Staff \relative c' {
  c16^( d e f
  \voiceOne
  <<
    { g4 f e | d2 e2) }
  \context Voice="1" { \voiceTwo
    r8 e4 d c8 ~ | c8 b16 a b8 g ~ g2
    \oneVoice
  }
  \new Voice { \voiceThree
    s2. s4 b4 c2
    \oneVoice
  }
  >>
  \oneVoice
}
```



Indem man den `\-`-Trenner vermeidet, gelingt es auch, mehrstimmige Abschnitte ineinander zu schachteln, was in manchen Fällen die bessere und natürlichere Lösung sein kann.

```
\new Staff \relative c' {
  c16^( d e f
  \voiceOne
  <<
    { g4 f e | d2 e2) }
  \context Voice="1" { \voiceTwo
    r8 e4 d c8 ~ |
    <<
      {c8 b16 a b8 g ~ g2}
      \new Voice { \voiceThree
        s4 b4 c2
        \oneVoice
      }
    >>
  \oneVoice
  }
  >>
  \oneVoice
}
```



In manchen Fällen von sehr komplexer polyphoner Musik können noch mehr Stimmen benötigt werden, um Zusammenstöße zwischen Noten zu vermeiden. Zusätzliche Stimmen werden durch einen neuen Bezeichner erstellt, wie das nächste Beispiel zeigt.

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)

\relative c''' <<
  { \voiceOne g4 ~ \stemDown g32[ f( es d c b a b64 )g] } \\\
  { \voiceThree b4} \\\
  { \voiceFive d,} \\\
  { \voiceTwo g,}
>>
```



## Auflösung von Zusammenstößen

Normalerweise werden Notenköpfe mit einer unterschiedlichen Anzahl von Punktierungen nicht verschmolzen, aber wenn die Objekt-Eigenschaft `merge-differently-dotted` in ein Notenkollisions ([Abschnitt "NoteCollision" in Programmreferenz](#))-Objekt gesetzt wird, werden sie zusammengefasst.

```
\new Voice << {
  g8 g8
  \override Staff.NoteCollision
    #'merge-differently-dotted = ##t
  g8 g8
} \\\ { g8.[ f16] g8.[ f16] } >>
```



Auf gleiche Art können auch Halbe mit Achteln vereinigt werden, indem `merge-differently-headed` eingesetzt wird:

```
\new Voice << {
  c8 c4.
  \override Staff.NoteCollision
    #'merge-differently-headed = ##t
  c8 c4. } \\\ { c2 c2 } >>
```



`merge-differently-headed` und `merge-differently-dotted` wirken sich allerdings nur auf Noten mit unterschiedlich gerichteten Hälsen aus (wie etwa Stimme 1 und 2).

LilyPond verschiebt auch Pausen vertikal, die einem Hals gegenüber stehen:

```
\new Voice << c''4 \\\ r4 >>
```



Wenn drei oder mehr Noten in der selben Spalte angeordnet werden, kann `merge-differently-headed` nicht mehr erfolgreich die Noten vereinen, die ineinander gesetzt werden müssten. Damit die Vereinigung funktioniert, muss der Befehl `\shift` vor die Note gesetzt werden, auf die er Auswirkung hat. Im ersten Takt des folgenden Beispiels funktioniert `merge-differently-headed` nicht (der Notenkopf der Halben ist schwarz). Im zweiten Takt wurde `\shiftOn` eingefügt, um das obere g aus der Spalte zu rücken, und das Vereinigen funktioniert wie gewünscht.

```
\override Staff.NoteCollision #'merge-differently-headed = ##t
<<
  { d='2 g2 } \
  { \oneVoice d='8 c8 r4 e,8 c'8 r4 } \
  { \voiceFour e,,2 e'2}
>>
<<
  { d='2 \shiftOn g2 } \
  { \oneVoice d='8 c8 r4 e,8 c'8 r4 } \
  { \voiceFour e,,2 e'2}
>>
```



## Vordefinierte Befehle

`\oneVoice`, `\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`.

`\shiftOn`, `\shiftOnn`, `\shiftOnnn`, `\shiftOff`: Diese Befehle definieren den Grad, mit welchem Noten der aktuellen Stimmen verschoben werden sollen. Die äußeren Stimmen (normalerweise 1 und 2) haben den Befehl `\shiftOff`, die inneren dagegen (drei und vier) den Befehl `\shiftOn`. `\shiftOnn` und `\shiftOnnn` stellen weitere Verschiebungsebenen dar.

Wenn LilyPond selber keine Lösung bieten kann, können die Eigenschaft `force-hshift` des Abschnitts *“NoteColumn”* in *Programmreferenz*-Objektes (siehe unten) sowie Pausen mit definierter Tonhöhe eingesetzt werden, um Satzentscheidungen des Programmes zu überschreiben.

```
\relative <<
{
  <d g>
  <d g>
} \ {
  <b f'>
  \once \override NoteColumn #'force-hshift = #1.7
  <b f'>
} >>
```



## Siehe auch

Programmreferenz: Die Objekte, die für Auflösung von Zusammenstößen zuständig sind, sind [Abschnitt “NoteCollision”](#) in *Programmreferenz* und [Abschnitt “RestCollision”](#) in *Programmreferenz*.

## Bekannte Probleme und Warnungen

Wenn `merge-differently-headed` mit einer nach oben gerichteten Achtel oder kleineren Note verwendet wird, und die nach unten gerichtete Note ist eine Halbe, bekommt die Achtel die falsche Richtung gesetzt.

Es gibt keine Unterstützung für Cluster, in denen die gleiche Note mit unterschiedlichen Vorzeichen im selben Akkord auftritt. In diesem Fall sollte man eine enharmonische Transkription benutzen oder die spezielle Cluster-Notationsweise, siehe [\[Cluster\]](#), [Seite 45](#).

## Automatische Kombination von Stimmen

### Musik parallel notieren

Noten für mehrere Stimmen können verschachtelt notiert werden:

```
\parallelMusic #'(voiceA voiceB) {
  r8      g'16[ c'' ] e''[ g' c'' e'' ] r8      g'16[ c'' ] e''[ g' c'' e'' ] |
  c'2                                c'2                                |
  r8      a'16[ d'' ] f''[ a' d'' f'' ] r8      a'16[ d'' ] f''[ a' d'' f'' ] |
  c'2                                c'2                                |
}
\new StaffGroup <<
  \new Staff \new Voice \voiceA
  \new Staff \new Voice \voiceB
>>
```



Das funktioniert ziemlich gut für Klaviernoten:

```
Noten = {
  \key c \major
  \time 4/4
```

```

\parallelMusic #'(voiceA voiceB voiceC voiceD) {
  % Takt 1
  r8 g'16[ c''] e''[ g' c'' e''] r8 g'16[ c''] e''[ g' c''
e''] |
  c'2                                c'2 |
  r8 a16[ d'] f'[ a d' f']          r8 a16[ d'] f'[ a d' f'] |
  c2                                c2 |

  % Takt 2
  a'8 b'          c'' d''          e'' f''          g'' a'' |
  d'4             d'             d'             d' |
  c16 d e f       d e f g       e f g a       f g a b |
  a,4             a,4             a,4             a,4 |

  % Takt 3 ...
}
}

\score {
  \new PianoStaff <<
    \Noten
    \new Staff <<
      \voiceA \
      \voiceB
    >>
    \new Staff {
      \clef bass
      <<
        \voiceC \
        \voiceD
      >>
    }
  >>
}

```



## 1.6 Notation innerhalb eines Systems



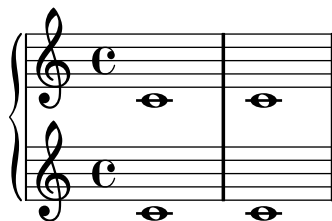
### 1.6.1 Systeme anzeigen lassen

#### Klammern am Systemanfang

Viele Partituren bestehen aus mehr als einem Notensystem. Diese Systeme können auf vier unterschiedliche Arten verbunden werden:

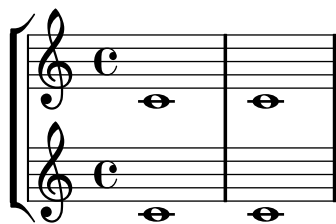
- Die Gruppe wird mit einer Klammer an der linken Seite geöffnet und die Taktlinien sind verbunden. Das ist der Klaviersystem ([Abschnitt “GrandStaff” in Programmreferenz](#))-Kontext.

```
\new GrandStaff
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



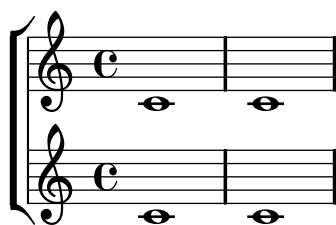
- Die Gruppe beginnt mit einer Klammer und die Taktlinien sind verbunden. Dieses Verhalten erzeugt der Stimmgruppen ([Abschnitt “StaffGroup” in Programmreferenz](#))-Kontext.

```
\new StaffGroup
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



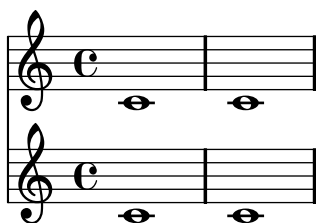
- Die Gruppe beginnt mit einer Klammer, aber die Taktlinien sind nicht miteinander verbunden. Das wird mit dem Chorsystem ([Abschnitt “ChoirStaff” in Programmreferenz](#))-Kontext erreicht.

```
\new ChoirStaff
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



- Die Gruppe beginnt mit einer vertikalen Linie. Taktlinien sind nicht verbunden. Das ist die Standardeinstellung für eine Partitur.

```
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



## Siehe auch

Die Definition der Taktlinien am Beginn jedes Systems werden mit den Befehlen [Abschnitt “SystemStartBar” in Programmreferenz](#), [Abschnitt “SystemStartBrace” in Programmreferenz](#) und [Abschnitt “SystemStartBracket” in Programmreferenz](#) festgelegt. Nur einer dieser drei Typen wird in jedem Kontext erstellt, und dieser Typ wird durch die `systemStartDelimiter`-Eigenschaft bestimmt.

## Übliche Veränderungen der Einstellungen

Anfangsklammern können tief einander verschachtelt werden.

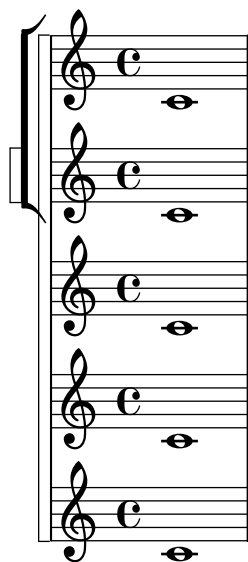
```
\new StaffGroup
\relative <<
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBracket a (SystemStartSquare b)) d)
```



```

\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
>>

```



## Das Notensystem

Noten, Dynamikzeichen usw. werden auf den Notenlinien angeordnet, die sich zu einem Notensystem zusammenfassen lassen. Das Programm LilyPond zeichnet diese Linien durch ein spezielles graphisches Objekt, **staff symbol** (engl. „staff“ = Notensystem) genannt.

Dieses Objekt kann bezüglich seiner Eigenschaften, wie Anzahl, Dicke und Abstand der Linien verändert werden.

Zusätzlich können Systeme beliebig begonnen und beendet werden. Das geschieht mit den Befehlen `\startStaff` und `\stopStaff`.

```

b4 b
\override Staff.StaffSymbol #'line-count = 2
\stopStaff \startStaff
b b
\revert Staff.StaffSymbol #'line-count
\stopStaff \startStaff
b b

```



Kombiniert mit verkleinerten Systemen, kann man diese Funktion etwa benutzen, um Ossia-Abschnitte zu notieren. Siehe das Beispiel:

## Siehe auch

Programmreferenz: [Abschnitt “StaffSymbol”](#) in *Programmreferenz*.

Beispiele: [Abschnitt “Staff notation”](#) in *Beispiele*.

## Systeme verstecken

## 1.6.2 Orchesterstimmen erstellen

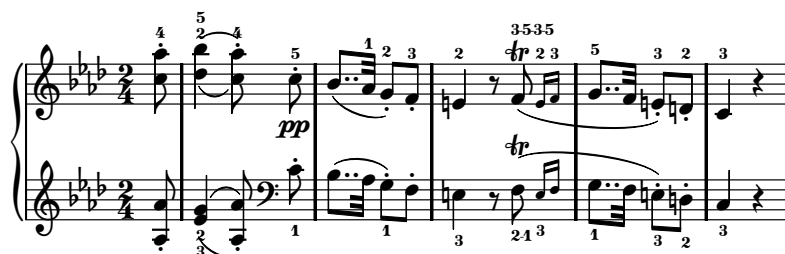
## Metronomangabe

## Instrumentenbezeichnung

## Stichnoten

## Stichnoten formatieren

## 1.7 Anmerkungen



## 1.7.1 Innerhalt des Systems

## Auswahl der Notations-Schriftgröße

## Fingersatzanweisungen

Fingersatzanweisungen können folgenderweise notiert werden:

*Note-Zahl*

Für Fingerwechsel muss eine Textbeschriftung (markup) benutzt werden:

c4-1 c-2 c-3 c-4

c<sup>^</sup>\markup { \finger "2 - 3" }



Mit dem Daumen-Befehl (`\thumb`) können die Noten bezeichnet werden, die mit dem Daumen (etwa auf dem Cello) gespielt werden sollen.

<a\_<sub>\thumb</sub> a'-3>8 <b\_<sub>\thumb</sub> b'-3>



Fingersätze für Akkorde können auch zu einzelnen Noten des Akkordes hinzugefügt werden, indem sie innerhalb der Akkord-Klammer direkt an die Noten angefügt werden.

```
< c-1 e-2 g-3 b-5 >4
```



## Übliche Veränderungen der Einstellungen

Eine bessere Kontrolle über die Position der Fingersätze in Akkorden lässt sich mit der Eigenschaft `fingeringOrientations` herstellen:

```
\set fingeringOrientations = #'(left down)
<c-1 es-2 g-4 bes-5 > 4
\set fingeringOrientations = #'(up right down)
<c-1 es-2 g-4 bes-5 > 4
```



Mit dieser Funktion können Fingersatzbezeichnungen auch bei einstimmiger Musik sehr nah in die Notenköpfe gerückt werden.

```
\set fingeringOrientations = #'(right)
<es'-2>4
```



## Siehe auch

Programmreferenz: [Abschnitt “Fingering”](#) in *Programmreferenz*.

Beispiele: [Abschnitt “Editorial annotations”](#) in *Beispiele*.

## Unsichtbare Noten

## Farbige Objekte

## Klammern

## Hälse

Immer, wenn das Programm eine Note findet, wird automatisch ein Notenhals ([Abschnitt “Stem”](#) in *Programmreferenz*) -Objekt erzeugt. Auch für ganze Noten und Pausen werden sie erzeugt, aber unsichtbar gemacht.

## Vordefinierte Befehle

`\stemUp` (Hälse nach oben), `\stemDown` (Hälse nach unten), `\stemNeutral` (Hälse je nach Notenposition).

## Übliche Veränderungen der Einstellungen

Um die Richtung der Hälse zu ändern, können die Befehle

```
a4 b c b
\override Stem #'neutral-direction = #up
a4 b c b
\override Stem #'neutral-direction = #down
a4 b c b
```



benutzt werden.

### 1.7.2 Außerhalb des Notensystems

#### Erklärungen in Ballonform

#### Gitternetzlinien

#### Analyseklammern

## 1.8 Text

### 1.8.1 Text eingeben

#### 1.8.1.1 Textarten

#### 1.8.1.2 Text mit Verbindungslinien

#### 1.8.1.3 Textartige Zeichen

#### 1.8.1.4 Separater Text

### 1.8.2 Text formatieren

#### 1.8.2.1 Textbeschriftung (Einleitung)

#### 1.8.2.2 Überblick über die wichtigsten Textbeschriftungsbefehle

#### 1.8.2.3 Textausrichtung

#### 1.8.2.4 Graphische Notation innerhalb einer Textbeschriftung

#### 1.8.2.5 Musikalische Notation innerhalb einer Textbeschriftung

#### 1.8.2.6 Textbeschriftung über mehrere Seiten

### **1.8.3 Schriftarten**

#### **1.8.3.1 Schriftart des gesamten Dokuments**

#### **1.8.3.2 Schriftarten für einen Eintrag**

## 2 Spezielle Notation

This chapter explains how to create musical notation for specific types of instrument or in specific styles.

### 2.1 Notation von Gesang

#### 2.1.1 Übliche Notation für Gesang

##### 2.1.1.1 Referenz für Musiktex te

##### 2.1.1.2 Einfache Lieder setzen

##### 2.1.1.3 Eingabe von Text

##### 2.1.1.4 Mit Gesangstexten und Bezeichnern arbeiten

#### 2.1.2 Text an einer Melodie ausrichten

##### 2.1.2.1 Automatische Silbendauer

##### 2.1.2.2 Manuelle Silbendauer

##### 2.1.2.3 Mehrere Silben zu einer Note

##### 2.1.2.4 Mehrere Noten zu einer Silbe

##### 2.1.2.5 Noten überspringen

##### 2.1.2.6 Unterstrichen und Trennstriche

##### 2.1.2.7 Gesangstext und Wiederholungen

#### 2.1.3 Positionierung von Gesangstext

##### 2.1.3.1 Getrennte Texte

##### 2.1.3.2 Text unabhängig von den Noten

##### 2.1.3.3 Hymnen

##### 2.1.3.4 Silben platzieren

##### 2.1.3.5 Gesangstext zwischen Systemen zentrieren

#### 2.1.4 Strophen

##### 2.1.4.1 Strophennummern hinzufügen

##### 2.1.4.2 Lautstärkebezeichnung zu Strophen hinzufügen

##### 2.1.4.3 Sängernamen zu Strophen hinzufügen

##### 2.1.4.4 Strophen mit unterschiedlichem Rhythmus

## Melismen ignorieren

## Zu einer alternativen Melodie umschalten

### 2.1.4.5 Die Strophen am Ende ausdrucken

### 2.1.4.6 Die Strophen am Ende in mehreren Spalten drucken

## 2.2 Tasteninstrumente

### 2.2.1 Übliche Notation für Tasteninstrumente

#### Referenz für Tasteninstrumente

Ein Pianosystem besteht aus zwei normalen Notensystemen, die durch eine Klammer verbunden sind. Die Systeme sind verhältnismäßig unabhängig, aber zeitweise können Stimmen zwischen den Systemen wechseln. Die gleiche Notation wird auch für andere Tasteninstrumente sowie Harfen verwendet. Das Klaviersystem (mit dem Befehl `PianoStaff` erstellt) ist speziell dazu geschaffen, Stimmen zwischen den Systemen wechseln zu lassen. In diesem Abschnitt werden das Klaviersystem und andere Eigenheiten der Pianonotation besprochen.

#### Bekannte Probleme und Warnungen

Dynamische Zeichen werden nicht automatisch mittig notiert, aber dafür gibt es Lösungen, vgl. die ‚piano centered dynamics‘-Vorlage in [Abschnitt „Piano templates“ in \*Handbuch zum Lernen\*](#).

#### Notensysteme manuell verändern

Stimmen können mit dem Befehl

```
\change Staff = Systembezeichnung Noten
```

manuell erzielt werden. Die Zeichenkette *Systembezeichnung* ist die Bezeichnung des Systems. Damit wird die aktuelle Stimme vom aktuellen System zu dem System mit der *Systembezeichnung* gewechselt. Üblicherweise ist die Systembezeichnung „up“ oder „down“. Dieses System (*Staff*), auf das verwiesen wird, muss auch wirklich existieren, weshalb die Aufstellung eines Klaviersystems dann üblicherweise damit beginnt, die Systeme zu benennen. Hierbei können natürlich beliebige Namen vergeben werden.

```
<<
\new Staff = "up" {
  \skip 1 * 10 % keep staff alive
}
\new Staff = "down" {
  \skip 1 * 10 % idem
}
>>
```

Die einzelnen Stimmen werden mit dem `Voice`-Befehl daraufhin eingefügt.

```
\context Staff = down
  \new Voice { ... \change Staff = up ... }
```

#### Automatischer Systemwechsel

Stimmen können angewiesen werden, automatisch zwischen dem oberen und unteren System zu wechseln. Die Syntax hierfür lautet:

```
\autochange ...Noten...
```

Damit werden zwei Notensysteme innerhalb des aktiven Klaviersystems erstellt, die „oben“ (up) und „unten“ (down) genannt werden. Auf dem unteren System wird als Standard der Bassschlüssel gesetzt.

Ein `\relative`-Abschnitt, der sich außerhalb des `\autochange`-Abschnittes befindet, hat keinen Einfluss auf die Notenhöhen der *Noten*. Es muss also bei Bedarf ein weiterer `\relative`-Befehl innerhalb des Abschnittes benutzt werden.

```
\autochange \relative ... ..
```

Der Wechsel wird automatisch basierend auf der Tonhöhe der Note vorgenommen (als Wechselfpunkt gilt das eingestrichene C). Dabei wird die Richtung auch über Pausen hinweg im Voraus bestimmt.

```
\new PianoStaff
\autochange \relative c'
{
  g4 a b c d r4 a g
}
```



## Siehe auch

Im Handbuch: [Abschnitt “Single staff”](#) in *Handbuch zum Lernen*.

Programmreferenz: [Abschnitt “AutoChangeMusic”](#) in *Programmreferenz*.

## Bekannte Probleme und Warnungen

Der Systemwechsel wird nicht immer an der optimalen Stelle vorgenommen. Für hohe Qualität muss der Wechsel mit der Hand eingegeben werden.

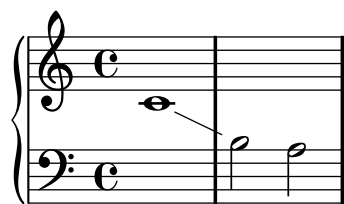
`\autochange` kann sich nicht innerhalb einer `\times`-Umgebung befinden.

## Stimmführungslinien

Immer, wenn eine Stimme von einem Klaviersystem zu dem anderen wechselt, kann automatisch eine Linie zur Verdeutlichung des Stimmenverlaufs ausgegeben werden. Das wird erreicht, indem `followVoice` aktiviert wird:

```
\new PianoStaff <<
\new Staff="one" {
  \set followVoice = ##t
  c1
  \change Staff=two
  b2 a
}
\new Staff="two" { \clef bass \skip 1*2 }
>>
```





## Siehe auch

Programmreferenz: [Abschnitt “VoiceFollower”](#) in *Programmreferenz*.

## Vordefinierte Befehle

`\showStaffSwitch`, `\hideStaffSwitch`.

## Hälsa über beide Systeme

Akkorde, die über zwei Systeme reichen, können erstellt werden, indem die Länge der Hälsa im unteren System vergrößert wird, bis sie zum oberen System hinauf reichen bzw. umgekehrt bei Hälsen, die nach unten zeigen.

```
stemExtend = {
  \once \override Stem #'length = #10
  \once \override Stem #'cross-staff = ##t
}
noFlag = \once \override Stem #'flag-style = #'no-flag
\new PianoStaff <<
  \new Staff {
    \stemDown \stemExtend
    f'4
    \stemExtend \noFlag
    f'8
  }
  \new Staff {
    \clef bass
    a4 a8
  }
>>
```



### 2.2.2 Klavier

#### Klavierpedal

Klaviere haben Pedale, mit denen die Klangeigenschaften beeinflusst werden können. Normalerweise besitzt ein Klavier drei Pedale, das linke oder Haltepedal, das rechte oder Una-corda-Pedal und das Sostenu-to-Pedal.

Spielanweisungen für das Pedal können durch die Befehle `\sustainOn` (linkes Pedal), `\sustainOff` (linkes Pedal hoch), `\unaCorda` (rechtes Pedal), `\treCorde` (rechtes Pedal hoch),

`\sostenutoOn` (mittleres Pedal) und `\sostenutoOff` (mittleres Pedal hoch) anschließend an eine Note oder einen Akkord notiert werden.

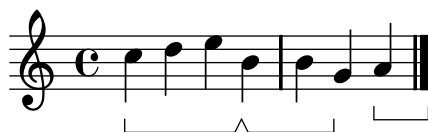
```
c'4\sustainOn c'4\sustainOff
```



Die Ausgabe kann verändert werden, indem der Wert What is printed can be modified by setting `pedalXStrings`, where *X* is one of the pedal types: `Sustain`, `Sostenuto` or `UnaCorda`. Refer to [Abschnitt “SustainPedal” in Programmreferenz](#) in the program reference for more information.

Pedalbenutzung kann auch durch eine Anzahl von Klammern angezeigt werden, wenn die `pedalSustainStyle`-Eigenschaft auf Klammern (eng. bracket) gesetzt wird.

```
\set Staff.pedalSustainStyle = #'bracket
c\sustainOn d e
b\sustainOff\sustainOn
b g \sustainOff a \sustainOn \bar "|."
```



Eine dritte Art der Pedalnotation besteht aus einer Mischung von Text und den Pedalklammern. Sie wird durch setzen von `pedalSustainStyle` auf `mixed` erreicht.

```
\set Staff.pedalSustainStyle = #'mixed
c\sustainOn d e
b\sustainOff\sustainOn
b g \sustainOff a \sustainOn \bar "|."
```



Die Einstellungen für linkes und mittleres Pedal ist üblicherweise ein  `#'text`-Stil. Für das mittlere Pedal ist der gemischte Stil voreingestellt.

```
c\sostenutoOn d e c, f g a\sostenutoOff
```



Die Erscheinung der Pedal-Klammer kann mit den Eigenschaften `edge-width`, `edge-height` und `shorten-pair` des `PianoPedalBracket`-Objekts verändert werden (siehe [Abschnitt “PianoPedalBracket” in Programmreferenz](#) in der Programmreferenz). Zum Beispiel kann die Klammer bis zum rechten Ende eines Notenkopfes verlängert werden.

```
\override Staff.PianoPedalBracket #'shorten-pair = #'(0 . -1.0)
c\sostenutoOn d e c, f g a\sostenutoOff
```



## Siehe auch

Im Handbuch: [\[Laissez-vibrer-Bögen\]](#), Seite 35

## 2.2.3 Akkordion

### Diskant-Symbole

Dieser Abschnitt wurde noch nicht übersetzt, bitte lesen Sie den Abschnitt im englischen Benutzerhandbuch.

## 2.3 Bundlose Saiteninstrumente

### 2.3.1 Übliche Notation für nichtgebundene Saiteninstrumente

#### 2.3.1.1 Hinweise für ungebundene Saiteninstrumente

#### 2.3.2 Streichinstrumente

##### 2.3.2.1 Referenz für Streicher

Flageolett-Töne werden mit einem anderen Notenkopfstil notiert. Sie werden eingegeben, indem die Flageolettnote mit dem Befehl `\harmonic` markiert wird.

```
<c g'\harmonic>4
```



#### 2.3.3 Zupfinstrumente

##### 2.3.3.1 Harfe

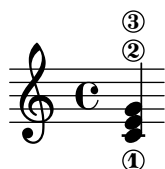
## 2.4 Saiteninstrumente mit Bündeln

### 2.4.1 Übliche Notation für gebundene Saiteninstrumente

#### Referenz für gebundene Saiteninstrumente

### Seitennummerbezeichnung

Saitennummerierung kann zu den Akkorden hinzugefügt werden, indem die Saitennummer nach einem Backslash als Zahl angegeben wird:



## Siehe auch

Programmreferenz: [Abschnitt “StringNumber” in Programmreferenz](#).

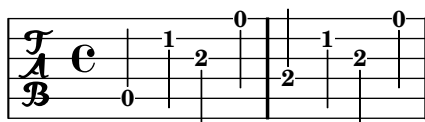
Beispiele: [Abschnitt “Fretted strings” in Beispiele](#).

## Standardtabulaturen

Tabulatur-Notation wird für die Notation von Zupfinstrumenten benutzt. Tonhöhen werden hier nicht durch Notenköpfe, sondern durch Zahlen notiert. Diese Zahlen zeigen an, auf welcher Saite und welchem Bund der Ton gespielt werden soll. LilyPond bringt beschränkte Unterstützung für Tabulaturen mit.

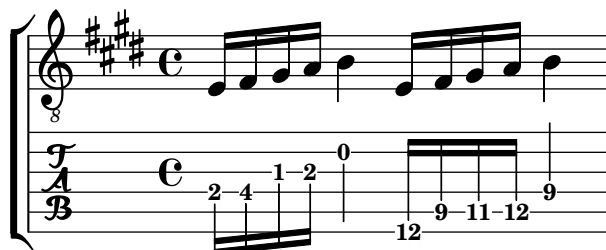
Die Saitennummer, die mit einer Note assoziiert ist, wird durch einen Backslash, gefolgt von einer Zahl, notiert, etwa `c4\3` für eine Viertelnote C auf der dritten Saite. In der Standardeinstellung ist die erste Saite die höchste Saite und als Stimmung wird die übliche Gitarrenstimmung auf sechs Saiten angenommen. Die Noten werden in einer Tabulatur gesetzt, indem [Abschnitt “TabStaff” in Programmreferenz](#) und [Abschnitt “TabVoice” in Programmreferenz](#)-Kontexte verwendet werden.

```
\new TabStaff {
  a,4\5 c'\2 a\3 e'\1
  e\4 c'\2 a\3 e'\1
}
```



Wenn keine Saite angegeben wird, wird die Saite, deren Bund-Zahl weniger als der Betrag in `minimumFret` beträgt, ausgewählt. Der Standardwert für `minimumFret` beträgt 0.

```
e16 fis gis a b4
\set TabStaff.minimumFret = #8
e16 fis gis a b4
```



## Übliche Veränderungen der Einstellungen

Um Tabulaturen mit Hälsen nach unten und horizontalen Balken zu erstellen, muss der `TabStaff` mit folgenden Parametern initialisiert werden:

```
\stemDown
\override Beam #'damping = #100000
```

## Siehe auch

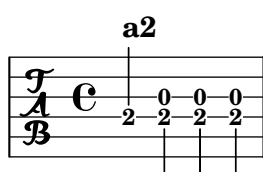
Programmreferenz: [Abschnitt “TabStaff” in Programmreferenz](#), [Abschnitt “TabVoice” in Programmreferenz](#).

## Bekannte Probleme und Warnungen

Akkorde werden nicht gesondert behandelt, sodass die Saitenauswahlfunktion eventuell die selbe Saite für zwei Töne eines Akkordes auswählen kann.

Damit die Kombination von Stimmen (`\partcombine`) richtig funktioniert, müssen speziell erstellte Stimmen innerhalb des Tabulatursystems (`TabStaff`) benutzt werden:

```
Melodie = \partcombine { e4 g g g }{ e4 e e e }
<<
  \new TabStaff <<
    \new TabVoice = "one" s1
    \new TabVoice = "two" s1
    \new TabVoice = "shared" s1
    \new TabVoice = "solo" s1
    { \Melodie }
  >>
>>
```

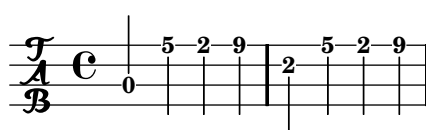


## Angepasste Tabaturen

Die Stimmung der Saiten kann verändert werden. Die Stimmung muss als eine Scheme-Liste dargestellt werden, in der jeder Saite eine Zahl zugewiesen ist, die die Tonhöhe der offenen Saite in Halbtönen (gemessen von c') darstellt. Die Liste wird als Definition von `stringTunings` dargestellt. Die Halbtöne können vom c' aus addiert oder subtrahiert werden. Die Reihenfolge der Zahlen bestimmt auch die Reihenfolge der Saiten. LilyPond entnimmt aus dieser Liste auch die Anzahl der Saiten.

Im nächsten Beispiel wird `stringTunings` für die Tonhöhen e, a, d, und g gesetzt.

```
\new TabStaff <<
  \set TabStaff.stringTunings = #'(-5 -10 -15 -20)
  {
    a,4 c' a e' e c' a e'
  }
>>
```



LilyPond bringt schon vordefinierte Stimmungen für Banjo, Mandoline, Gitarre und Bassgitarre mit.

```
\set TabStaff.stringTunings = #bass-tuning
```

Der Standard ist die Gitarrenstimmung (`guitar-tuning`) mit den Saiten EADGBE. Andere vordefinierte Stimmungen sind `guitar-open-g-tuning`, `mandolin-tuning` und `banjo-open-g-tuning`.

## Siehe auch

In der Datei ‘`scm/output-lib.scm`’ sind die Stimmungen definiert. Programmreferenz: [Abschnitt “`Tab\_note\_heads\_engraver`” in \*Programmreferenz\*](#).

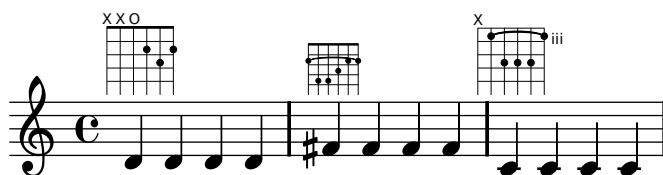
## Bekannte Probleme und Warnungen

Spezialeffekte für Gitarren sind noch nicht implementiert.

## Bund-Diagramm-Beschriftung

Akkordtabulaturen (Fret-Diagramme) können zu Noten als eine Beschriftung hinzugefügt werden. Die Beschriftung enthält Information über die erwünschte Akkordtabulatur, wie das folgende Beispiel zeigt:

```
\new Voice {
  d'\markup \fret-diagram #"6-x;5-x;4-o;3-2;2-3;1-2;"
  d' d' d'
  fis'\markup \override #'(size . 0.75) {
    \override #'(finger-code . below-string) {
      \fret-diagram-verbose #'((place-fret 6 2 1) (barre 6 1 2)
                             (place-fret 5 4 3) (place-fret 4 4 4)
                             (place-fret 3 3 2) (place-fret 2 2 1)
                             (place-fret 1 2 1))
    }
  }
  fis' fis' fis'
  c'\markup \override #'(dot-radius . 0.35) {
    \override #'(finger-code . in-dot) {
      \override #'(dot-color . white) {
        \fret-diagram-terse #"x;3-1-(;5-2;5-3;5-4;3-1-);"
      }
    }
  }
  c' c' c'
}
```



Es gibt drei Modi für die Abbildung der Tabulatur: den Standardmodus, einen ausführlichen Modus und einen knappen Modus. Die unterschiedlichen Modi werden angewählt, indem an den Befehl `fret-diagram` die Begriffe `-standard`, `-verbose` bzw. `-terse` angehängt werden. Die Modi produzieren gleichwertige Ausgaben, haben aber unterschiedliche Informationen im Beschriftungsbefehl. Details über die Beschriftungsbefehle finden sich im Abschnitt

**Abschnitt B.6 [Text markup commands], Seite 135.** Alle drei Modi werden im obigen Beispiel gezeigt.

Einige graphische Eigenschaften können verändert werden. Genauer zur Schnittstelle für die Griffsymboleigenschaften findet sich unter **Abschnitt “fret-diagram-interface” in Programmreferenz.**

## Siehe auch

Beispiele: **Abschnitt “Fretted strings” in Beispiele.**

## Fingersatz der rechten Hand

Fingersatz für die rechte Hand in Akkorden kann notiert werden, indem der Befehl `note-\rightHandFinger finger` benutzt wird.

```
<c-\rightHandFinger #1 e-\rightHandFinger #2 >
```



Zur Erleichterung kann der Befehl `\rightHandFinger` zu ein paar Buchstaben abgekürzt werden, etwa RH.

```
#(define RH rightHandFinger)
```

## Übliche Veränderungen der Einstellungen

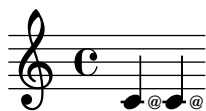
Größere Kontrolle über den Fingersatz der rechten Hand kann mit einer Definition des Befehls `strokeFingerOrientations` erreicht werden:

```
#(define RH rightHandFinger)
{
  \set strokeFingerOrientations = #'(up down)
  <c-\RH #1 es-\RH #2 g-\RH #4 > 4
  \set strokeFingerOrientations = #'(up right down)
  <c-\RH #1 es-\RH #2 g-\RH #4 > 4
}
```



Die Buchstaben, die für den Fingersatz verwendet werden, sind in der Eigenschaft `digit-names` enthalten, aber sie können auch individuell verändert werden, indem `\rightHandFinger` mit einer Zeichenfolge als Argument eingesetzt wird, wie im nächsten Beispiel:

```
#(define RH rightHandFinger)
{
  \set strokeFingerOrientations = #'(right)
  \override StrokeFinger #'digit-names = #'("x" "y" "z" "!" "@")
  <c-\RH #5 >4
  <c-\RH "@">4
}
```



## Siehe auch

Program reference: [Abschnitt “StrokeFinger”](#) in *Programmreferenz*

### 2.4.2 Gitarre

#### Position und Barret anzeigen

Dieses Beispiel zeigt, wie Griff - und Barre-Angaben eingefügt werden können.

```
\clef "G_8"
b16 d16 g16 b16 e16
\textSpannerDown
\override TextSpanner #'bound-details #'left #'text = #"XII "
  g16\startTextSpan
    b16 e16 g16 e16 b16 g16\stopTextSpan
e16 b16 g16 d16
```



Gedämpfte (X) Notenköpfe werden in der Gitarrenmusik benutzt, um zu signalisieren, dass der Gitarrist eine Note oder einen Akkord spielen soll, indem seine Finger die Saiten nur berühren, anstatt sie vollständig herunterzudrücken. Damit werden schlagzeugartige Geräusche erreicht, die aber noch die ursprüngliche Tonhöhe ahnen lassen. Diese Spielart wird mit Kreuz-Notenköpfen notiert, wie im Abschnitt [\[Besondere Notenköpfe\]](#), [Seite 10](#) gezeigt.

#### Flageolett und gedämpfte Noten

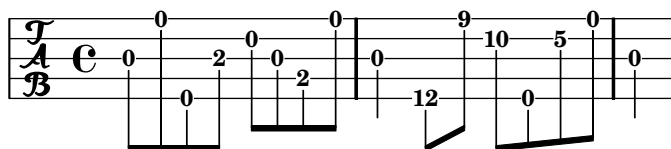
### 2.4.3 Banjo

#### Banjo-Tabulaturen

LilyPond unterstützt die Basisnotation für das fünfsaitige Banjo. Die Banjo-Tabulatur-Funktion sollte zum Notieren von Banjo-Tabulaturen verwendet werden, damit die richtigen Bund-Nummern für die fünfte Saite gesetzt werden:

```
\new TabStaff <<
  \set TabStaff.tablatureFormat = #fret-number-tablature-format-banjo
  \set TabStaff.stringTunings = #banjo-open-g-tuning
  {
    \stemDown
    g8 d' g'\5 a b g e d' |
    g4 d''8\5 b' a'\2 g'\5 e'\2 d' |
    g4
  }
>>
```





Eine Anzahl von üblichen Stimmungen für das Banjo sind in LilyPond schon vordefiniert: `banjo-c-tuning` (gCGBD), `banjo-modal-tuning` (gDGCD), `banjo-open-d-tuning` (aDF#AD) und `banjo-open-dm-tuning` (aDFAD).

Diese Stimmungen können für das viersaitige Banjo angepasst werden, indem die `four-string-banjo`-Funktion eingesetzt wird:

```
\set TabStaff.stringTunings = #(four-string-banjo banjo-c-tuning)
```

## Siehe auch

Die Datei `'scm/output-lib.scm'` enthält vordefinierte Banjo-Stimmungen.

## 2.5 Schlagzeug

### 2.5.1 Übliche Notation für Schlagzeug

Rhythmusnotation wird vor allem für Schlaginstrumente eingesetzt, aber hiermit kann auch der Rhythmus einer Melodie dargestellt werden.

#### 2.5.1.1 Referenz für Schlagzeug

#### 2.5.1.2 Grundlagen der Schlagzeugnotation

Schlagzeug-Noten können im `\drummode`-Modus notiert werden, der sich ähnlich verhält wie der Standardmodus für die Noteneingabe. Jedes Schlagzeuginstrument hat einen langen Namen und eine Abkürzung, und beide können nebeneinander benutzt werden.

```
\drums {
  hihat hh bassdrum bd
}
```



Eine vollständige Liste der Schlagwerk-Bezeichnungen findet sich in der Datei `'ly/drumpitch-init.ly'`.

## Siehe auch

Programmreferenz: [Abschnitt "note-event" in Programmreferenz](#).

### 2.5.1.3 Schlagzeugsysteme

Ein Schlagzeug-System besteht üblicherweise aus einem Notensystem mit mehreren Linien, wobei jede Linie ein bestimmtes Schlagzeug-Instrument darstellt.

Um die Noten darstellen zu können, müssen sie sich innerhalb von einem [Abschnitt "Drum-Staff" in Programmreferenz](#)- und einem [Abschnitt "DrumVoice" in Programmreferenz](#)-Kontext befinden.

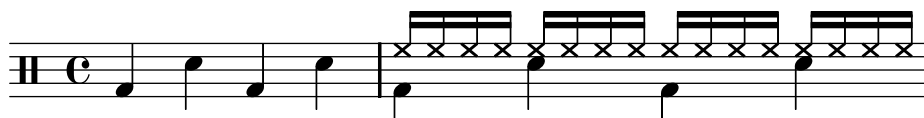
```
oben = \drummode { crashcymbal4 hihat8 halfopenhihat hh hh hh openhihat }
unten = \drummode { bassdrum4 snare8 bd r bd sn4 }
```

```
\new DrumStaff <<
  \new DrumVoice { \voiceOne \oben }
  \new DrumVoice { \voiceTwo \unten }
>>
```



Das Beispiel zeigt ausdrücklich definierte mehrstimmige Notation. Die Kurznotation für mehrstimmige Musik, wie sie im Abschnitt [\[Einfache Mehrstimmigkeit\]](#), Seite 45 beschrieben wird, kann auch verwendet werden, wenn die Abschnitt “DrumVoice” in *Programmreferenz* (Schlagzeugstimmen) am Anfang explizit initialisiert werden.

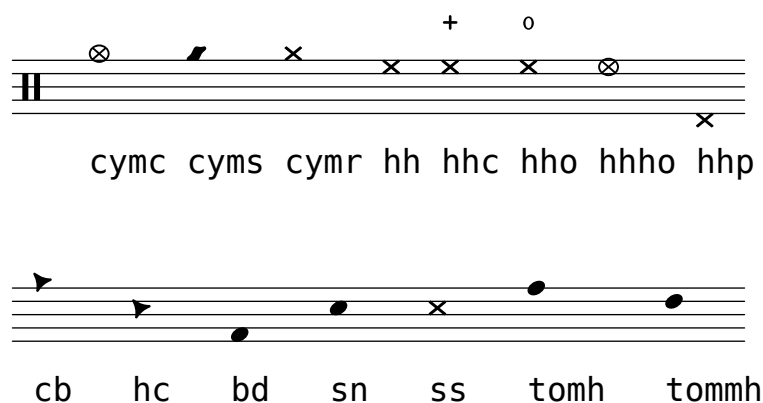
```
\new DrumStaff <<
  \new DrumVoice = "1" { s1 *2 }
  \new DrumVoice = "2" { s1 *2 }
  \drummode {
    bd4 sn4 bd4 sn4
    <<
      { \repeat unfold 16 hh16 }
      \\
      { bd4 sn4 bd4 sn4 }
    >>
  }
>>
```

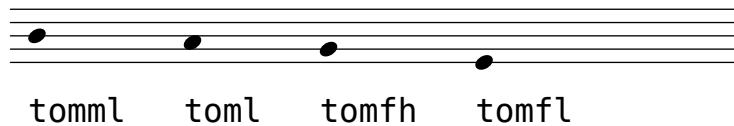


Es gibt auch weitere Layout-Einstellungen. Um diese zu verwenden, muss die Eigenschaft `drumStyleTable` im Abschnitt “DrumVoice” in *Programmreferenz*-Kontext entsprechend eingestellt werden. Folgende Variablen sind vordefiniert:

#### drums-style

Das ist die Standardeinstellung. Hiermit wird ein typisches Schlagzeug-System auf fünf Notenlinien erstellt.

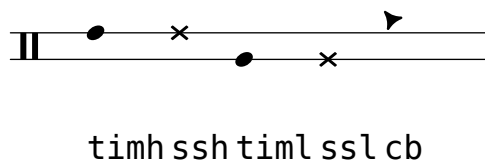




Die Schlagzeugdefinitionen unterstützen sechs unterschiedliche Tom Toms. Falls eine geringere Anzahl verwendet wird, kann man einfach die Tom Toms auswählen, deren Notation man haben will. Tom Toms auf den drei mittleren Linien werden mit den Bezeichnungen `tommh`, `tomml` und `tomfh` notiert.

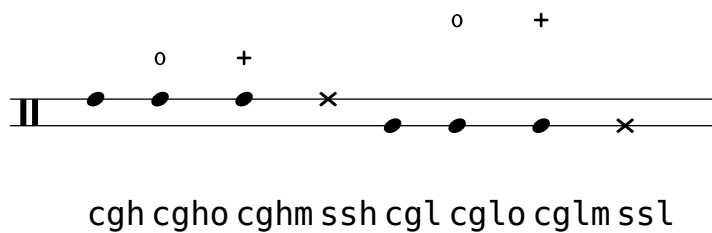
#### timbales-style

Hiermit werden Timbale auf zwei Notenlinien gesetzt.



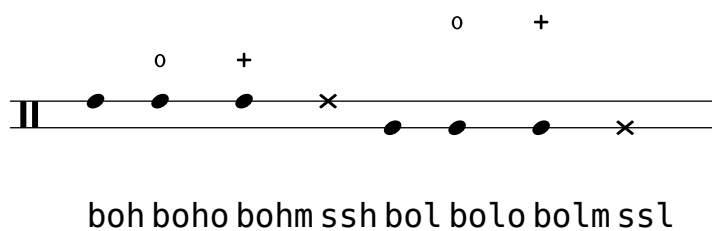
#### congas-style

Hiermit werden Congas auf zwei Linien gesetzt.



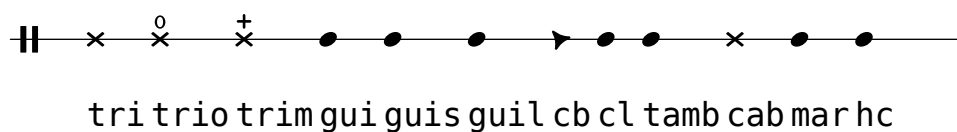
#### bongos-style

Hiermit werden Bongos auf zwei Linien gesetzt.



#### percussion-style

Dieser Stil ist für alle einfachen Perkussionsinstrumente auf einer Notenlinie.



Wenn ihnen keine der vordefinierten Stile gefällt, können Sie auch eine eigene Liste der Positionen und Notenköpfe am Anfang ihrer Datei erstellen.

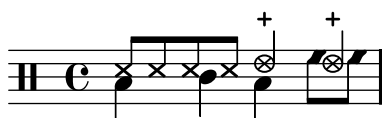
```
#(define mydrums '(
  (bassdrum default #f -1)
  (snare default #f 0)
```

```

      (hihat      cross      #f      1)
      (pedalhihat xcircle    "stopped" 2)
      (lowtom     diamond    #f      3)))
oben = \drummode { hh8 hh hh hh hhp4 hhp }
unten = \drummode { bd4 sn bd toml8 toml }

\new DrumStaff <<
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \new DrumVoice { \voiceOne \oben }
  \new DrumVoice { \voiceTwo \unten }
>>

```



## Siehe auch

Init-Dateien: 'ly/drumpitch-init.ly'.

Programmreferenz: [Abschnitt "DrumStaff"](#) in *Programmreferenz*, [Abschnitt "DrumVoice"](#) in *Programmreferenz*.

## Bekannte Probleme und Warnungen

Weil in den allgemeinen MIDI-Definitionen kein Rimshot enthalten ist, wird sidestick für diesen Zweck benutzt.

### 2.5.1.4 Geisternoten

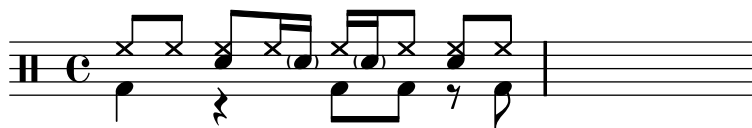
Geisternoten für Schlagzeug und Perkussion können mit dem Klammer- (`\parenthesize`)-Befehl, beschrieben in [\[Klammern\]](#), [Seite 57](#), erstellt werden. Im Standard-`\drummode`-Modus ist aber das `Parenthesis_engraver`-Plugin nicht automatisch enthalten. Sie müssen das Plugin ausdrücklich in den Kontext-Definitionen laden, wie im [Abschnitt "Modifying context properties"](#) in *Handbuch zum Lernen* detailliert beschrieben.

```

\new DrumStaff \with {
  \consists "Parenthesis_engraver"
} <<
  \context DrumVoice = "1" { s1 *2 }
  \context DrumVoice = "2" { s1 *2 }
  \drummode {
    <<
      {
        hh8[ hh] <hh sn> hh16
        < \parenthesize sn > hh < \parenthesize
        sn > hh8 <hh sn> hh
      } \ {
        bd4 r4 bd8 bd r8 bd
      }
    >>
  }
}

```

&gt;&gt;



Um jede Klammer-Definition (`\parenthesize`) müssen zusätzlich die spitzen Klammern für Akkorde (`< >`) gesetzt werden.

## 2.6 Blasinstrumente

### 2.6.1 Übliche Notation für Bläser

#### Referenz für Blasinstrumente

#### Fingersatz

### 2.6.2 Dudelsack

#### Dudelsack-Defintionen

LilyPond besitzt spezielle Definitionen, mit der die Musik des schottischen Hochland-Dudelsacks notiert wird. Um sie zu benutzen, muss

```
\include "bagpipe.ly"
```

am Anfang der LilyPond-Quelldatei eingefügt werden. Hierdurch können dann bestimmte Verzierungsnoten, die für die Dudelsackmusik üblich sind, mit kurzen Befehlen eingefügt werden. So reicht etwa der Befehl `\taor`, anstatt

```
\grace { \small G32[ d G e] }
```

zu schreiben.

`bagpipe.ly` enthält außerdem Definitionen für Tonhöhen von Dudelsacknoten in bestimmten Oktaven, so dass man sich nicht mehr um `\relative` oder `\transpose` kümmern muss.

```
\include "bagpipe.ly"
```

```
{ \grg G4 \grg a \grg b \grg c \grg d \grg e \grg f \grA g A }
```



Musik für den Dudelsack wird dem Namen nach in D-Dur geschrieben (auch wenn das eigentlich nicht stimmt). Weil das aber die einzige Tonart ist, die benutzt werden kann, werden die Vorzeichen meistens nicht geschrieben. Damit das funktioniert, müssen die Noten immer mit `\hideKeySignature` beginnen. Wenn die Vorzeichen hingegen angezeigt werden sollen, kann das mithilfe des Befehls `\showKeySignature` vorgenommen werden.

Some modern music use cross fingering on c and f to flatten those notes. This can be indicated by `cflat` or `fflat`. Similarly, the piobaireachd high g can be written `gflat` when it occurs in light music.

## Dudelsack-Beispiele

So sieht die bekannte Melodie Amazing Grace aus, wenn man sie für Dudelsack notiert.

```
\include "bagpipe.ly"
\layout {
  indent = 0.0\cm
  \context { \Score \remove "Bar_number_engraver" }
}

\header {
  title = "Amazing Grace"
  meter = "Hymn"
  arranger = "Trad. arr."
}

{
  \hideKeySignature
  \time 3/4
  \grg \partial 4 a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg e8. f16
  \dblA A2 \grg A4
  \grg A2 f8. A16
  \grg A2 \hdbl f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 e4
  \thrwd d2.
  \slurd d2
  \bar "|."
}
```

## Amazing Grace

Hymn

Trad. arr.





## 2.7 Notation von Akkorden

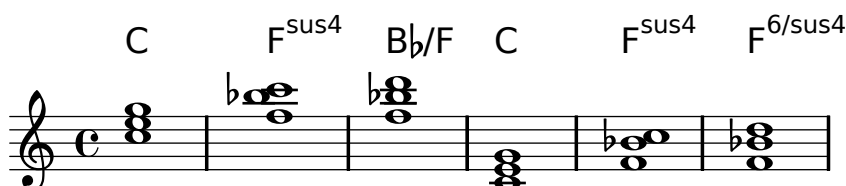
### 2.7.1 Akkord-Modus

#### Überblick über den Akkord-Modus

LilyPond bringt Unterstützung zur Ausgabe von Akkordsymbolen mit. Akkorde können entweder mithilfe der musikalischen Akkordeingabe (also `< ... >`) notiert werden, aber sie können auch mit ihren Bezeichnungen direkt eingegeben werden. Programmintern werden die Akkorde als eine Gruppe von Tonhöhen dargestellt und können somit transponiert werden.

```
twoWays = \transpose c c' {
  \chordmode {
    c1 f:sus4 bes/f
  }
  <c e g>
  <f bes c'>
  <f bes d'>
}
```

```
<< \new ChordNames \twoWays
  \new Voice \twoWays >>
```



Dieses Beispiel zeigt auch, dass die Ausgaberroutinen für die Akkordsymbole nicht besonders klug sind. Der letzte Akkord (`f bes d`) wird nicht als Umkehrung erkannt.

Achtung: Die Dauer eines Akkordes muss außerhalb der spitzen Klammern `<>` notiert werden.

```
<c e g>2
```

#### Übliche Akkorde

Im Akkord-Modus wird eine Anzahl von Tonhöhen (ein Akkord) durch eine normale Notenbezeichnung notiert. Ein Akkord wird mit seinem Grundton notiert, der einfach wie eine normale Note geschrieben wird:

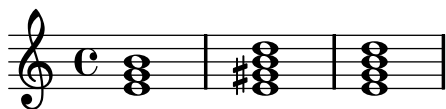
```
\chordmode { es4. d8 c2 }
```



Dieser Modus wird mit dem `\chordmode`-Befehl aktiviert.

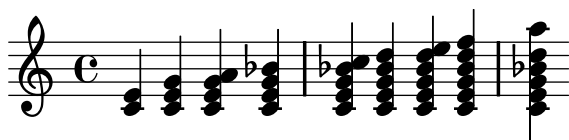
Andere als die Grundakkorde können notiert werden, indem an die Notenbezeichnung ein Doppelpunkt und eine Spezifizierung (die auch aus einer Zahl bestehen kann) angefügt wird.

```
\chordmode { e1:m e1:7 e1:m7 }
```



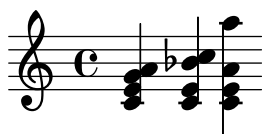
Die erste Zahl, die auf den Grundton folgt, wird als „Art“ des Akkordes interpretiert: Terzen werden auf dem Grundton gestapelt, bis die angegebene Zahl (=Tonhöhe) erreicht ist. Eine Ausnahme ist c:13, weil hier die 11 ausgelassen wird.

```
\chordmode { c:3 c:5 c:6 c:7 c:8 c:9 c:10 c:11 c:13 }
```



Kompliziertere Akkorde können auch konstruiert werden, indem einzelne Intervalle zu dem Grundton addiert werden. Diese Additionen werden nach der ersten Zahl notiert und mit Punkten voneinander getrennt.

```
\chordmode { c:5.6 c:3.7.8 c:3.6.13 }
```



Einzelne Intervalle können mit - oder + vergrößert oder verkleinert werden.

```
\chordmode { c:7+ c:5+.3- c:3-.5-.7- }
```



Zu entfernende Töne werden mit der gleichen Methode notiert, allerdings mit einem Dach (^) vor der Sequenz, die nicht erscheinen soll. Sie müssen nach den zu addierenden Tönen notiert werden.

```
\chordmode { c^3 c:7^5 c:9^3.5 }
```



Mit weiteren Endungen können Tonhöhen der Akkorde beeinflusst werden. Folgende Endungen sind unterstützt:

- |     |   |
|-----|---|
| m   | Der Moll-Akkord. Diese Endung erniedrigt die Terz (3) und die Septime (7), wenn vorhanden.                |
| dim | Der verminderte Akkord. Diese Endung erniedrigt die Terz (3), Quinte (5) und Septime (7), wenn vorhanden. |



- aug** Der übermäßige Akkord. Diese Endung erhöht die Quinte (5).  
**maj** Der große Sept-Akkord. Diese Endung erhöht die Septime (7), wenn vorhanden.  
**sus** Der Sekundakkord. Diese Endung entfernt die Terz (3) aus dem Akkord. Schreiben Sie 2 oder 4 hinter die Endung, um entweder die Sekunde (2) oder die Quarte (4) zum Akkord hinzuzufügen.

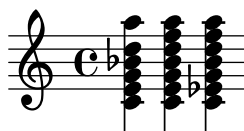
Die Endungen können mit weiteren hinzugefügten Tönen kombiniert werden.

```
\chordmode { c:sus4 c:7sus4 c:dim7 c:m6 }
```



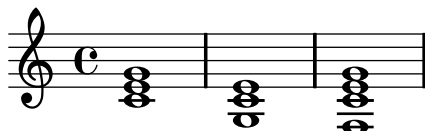
Da der unveränderte elfte Ton in Verbindung mit einem unveränderten 13. Ton nicht gut klingt, wird die 11 in diesem Fall automatisch entfernt (es sei denn, sie wird explizit gefordert).

```
\chordmode { c:13 c:13.11 c:m13 }
```



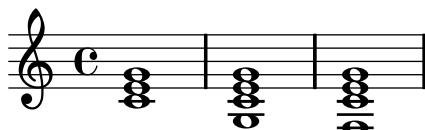
Eine Umkehrung (ein Ton des Akkordes wird unter den Grundton gesetzt) sowie auch Bassnoten können mit dem Schrägstrich (/) markiert werden:

```
\chordmode { c1 c/g c/f }
```



Eine Bassnote kann hinzugefügt werden, anstatt dass sie aus dem Akkord entnommen wird, indem noch ein Plus zwischen den Schrägstrich und die Tonhöhe gesetzt wird:

```
\chordmode { c1 c/+g c/+f }
```



Der Akkord-Modus funktioniert ähnlich wie der Liedtext-Modus (`\lyricmode`) usw. Die meisten Befehle funktionieren weiter, so können etwa `r` und `\skip` benutzt werden, um Pausen und Zwischenräume einzufügen. Mit Eigenschaftsbefehlen können verschiedene Einstellungen verändert werden.

## Bekannte Probleme und Warnungen

Jedes Intervall kann nur einmal in einem Akkord vorkommen. Im folgenden Beispiel wird nur der erweiterte Akkord erstellt, weil 5+ als letztes kommt.

```
\chordmode { c:5.5-.5+ }
```



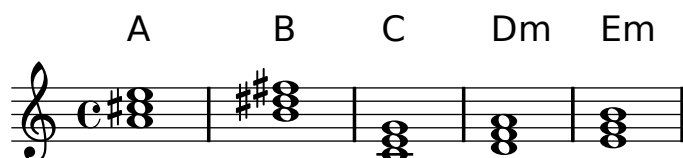
## Erweiterte und modifizierte Akkorde

### 2.7.2 Akkorde anzeigen

#### Akkordbezeichnungen drucken

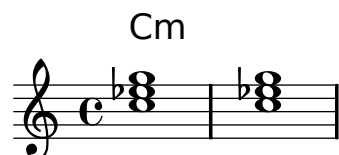
Um Akkordsymbole anstelle der Noten zu setzen, muss der Abschnitt “ChordNames” in *Programmreferenz*-Kontext. Die Akkorde können entweder mit der Notation aus dem vorigen Abschnitt oder direkt mit der Akkordeingabe für Noten (< und >) notiert werden.

```
harmonies = {
  \chordmode {a1 b c} <d' f' a'> <e' g' b'>
}
<<
  \new ChordNames \harmonies
  \new Staff \harmonies
>>
```



Sie können Akkordwiederholungen unterdrücken, so dass nur bei einer Änderung des Akkordes ein Symbol ausgegeben wird. Das aktuelle Symbol wird auch am Zeilenanfang wiederholt. Das geschieht mit dem Befehl `\set chordChanges = ##t`.

```
harmonies = \chordmode {
  c1:m c:m \break c:m c:m d
}
<<
  \new ChordNames {
    \set chordChanges = ##t
    \harmonies }
  \new Staff \transpose c c' \harmonies
>>
```

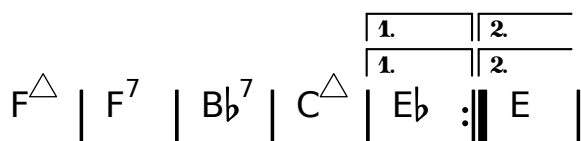


In den vorangegangenen Beispielen wurden alle Akkorde über dem Notensystem mit den entsprechenden Noten ausgegeben. Das ist aber nicht erforderlich. Akkorde können auch alleine gedruckt werden. Dabei kann es nötig sein, Abschnitt “Volta\_engraver” in *Programmreferenz* und Abschnitt “Bar\_engraver” in *Programmreferenz* einzufügen, damit Wiederholungen angezeigt werden.

```

\new ChordNames \with {
  \override BarLine #'bar-size = #4
  \consists Bar_engraver
  \consists "Volta_engraver"
}
\chordmode { \repeat volta 2 {
  f1:maj7 f:7 bes:7
  c:maj7
} \alternative {
  es e
}
}

```



Die Standardeinstellungen für die Symbole entsprechen den Konventionen im Jazz, wie sie von Klaus Ignatzek (siehe [Anhang A \[Literatur\]](#), Seite 116). vorgeschlagen wurden. Mit den folgenden Befehlen könne aber Anpassungen vorgenommen werden:

#### chordNameExceptions

Das ist eine Liste, die alle Akkorde aufführt, die ungewöhnliche Form haben.

Die Liste muss nach dem Schema

```
{ <c f g bes>1 \markup { \super "7" "wahn" } }
```

gesetzt werden.

Diese Liste in die `chordNameExceptions`-Funktion einzulesen ist allerdings etwas kompliziert. Mit dem folgenden Code-Abschnitt wird `chExceptionMusic` (aufeinander folgende Noten) in eine Ausnahmeliste umgewandelt:

```
(sequential-music-to-chord-exceptions chExceptionMusic #t)
```

Dann werden mit dem Code

```
(append
```

```
  (sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)
```

die neuen Ausnahmen zu den schon vorhandenen hinzugefügt, die in der Datei `'ly/chord-modifier-init.ly'` definiert sind.

*Chord name exceptions*

The property `chordNameExceptions` can be used to store a list of special notations for specific chords.

```
% modify maj9 and 6(add9)
```

```
% Exception music is chords with markups
```

```
chExceptionMusic = {
```

```
  <c e g b d'>1-\markup { \super "maj9" }
```

```
  <c e g a d'>1-\markup { \super "6(add9)" }
```

```
}
```

```
% Convert music to list and prepend to existing exceptions.
```

```
chExceptions = #( append
```

```
  ( sequential-music-to-chord-exceptions chExceptionMusic #t)
```

```

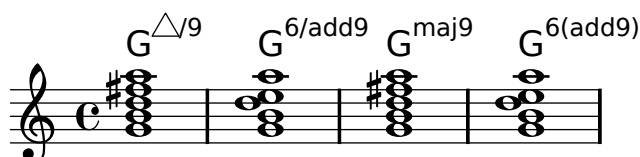
    ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

\layout {
  ragged-right = ##t
}

<< \context ChordNames \theMusic
    \context Voice \theMusic
>>

```



#### majorSevenSymbol

Mit dieser Eigenschaft wird das Aussehen der Notation für die große Septime (7) bestimmt. Vordefiniert sind die Optionen `whiteTriangleMarkup` und `blackTriangleMarkup`.

#### chordNameSeparator

Verschiedene Teile eines Akkordsymbolen werden normalerweise durch einen Schrägstrich markiert. Indem `chordNameSeparator` ein anderer Wert zugewiesen wird, kann das Aussehen des Trenners verändert werden, z. B.:

```

\new ChordNames \chordmode {
  c:7sus4
  \set chordNameSeparator
    = \markup { \typewriter "|" }
  c:7sus4
}

```

$C^{7/sus4}$   $C^7|sus4$

#### chordRootNamer

Der Grundton eines Akkordes wird normalerweise als Buchstabe mit optionaler Erhöhung oder Erniedrigung dargestellt. Die Interpretation einer Tonhöhe als Buchstabe wird von der `chordRootNamer`-Eigenschaft übernommen. Besondere Bezeichnungen, wie etwa im Deutschen H für einen H-Dur-Akkord (und nicht „B“ wie im Englischen), können durch Hinzufügen einer neuen Funktion zu dieser Eigenschaft erstellt werden.

#### chordNoteNamer

In den Standardeinstellungen wird ein einziger Buchstabe für einen Akkord ausgegeben, nämlich sein Grundton, der von der `chordRootNamer`-Eigenschaft bestimmt wird. Die `chordNoteNamer`-Eigenschaft hingegen kann das Aussehen dieses Buchstaben ändern, ihn etwa als Klein- und nicht als Großbuchstabe darstellen.

**chordPrefixSpacer**

Das „m“ für Moll-Akkorde wird normalerweise direkt hinter dem Akkordbuchstaben gesetzt. Mit der Eigenschaft `chordPrefixSpacer` kann ein Abstand(halter) zwischen den Buchstaben und das „m“ gesetzt werden. Der Abstandhalter wird nicht verwendet, wenn der Grundton erhöht oder erniedrigt ist.

Die vordefinierten Befehle `\germanChords`, `\semiGermanChords`, `\italianChords` und `\frenchChords` benutzen diese Variablen. Die Auswirkungen werden im nächsten Beispiel gezeigt.

default	E/D	Cm	B/B	B <sup>#</sup> /B <sup>#</sup>	B <sup>b</sup> /B <sup>b</sup>
german	E/d	Cm	H/h	H <sup>#</sup> /his	B/b
semi-german	E/d	Cm	H/h	H <sup>#</sup> /his	B <sup>b</sup> /b
italian	Mi/Re	Do m	Si/Si	Si <sup>#</sup> /Si <sup>#</sup>	Si <sup>b</sup> /Si <sup>b</sup>
french	Mi/Ré	Do m	Si/Si	Si <sup>#</sup> /Si <sup>#</sup>	Si <sup>b</sup> /Si <sup>b</sup>



Zusätzlich sind zwei weitere Vorlagen: eine alternative Jazz-Notation und eine systematische Notation, genannt Banter-Akkorde. Die alternative Jazz-Notation findet sich auch in der Liste im Abschnitt [Abschnitt B.1 \[Liste der Akkordbezeichnungen\]](#), Seite 117.

**Vordefinierte Befehle**

`\germanChords`, `\semiGermanChords`, `\italianChords`, `\frenchChords`.

**Siehe auch**

Beispiele: [Abschnitt “Chords” in Beispiele](#).

Init-Dateien: ‘scm/chords-ignatzek.scm’ und ‘scm/chord-entry.scm’.

**Bekannte Probleme und Warnungen**

Akkordsymbole werden nur von den Tonhöhenbezeichnungen bestimmt. Akkordumkehrungen sind nicht definiert, noch werden Bassnoten hinzugefügt. Deshalb kann es zu seltsamen Akkord-Symbolen kommen, wenn die Akkorde mit der `< ... >` Syntax notiert werden.

**Akkordbezeichnungen anpassen****2.7.3 Generalbass****Grundlagen des Bezifferten Basses**

LilyPond stellt Unterstützung für Generalbassnotation zur Verfügung.

```
<<
\new Voice { \clef bass dis4 c d ais g fis}
\new FiguredBass \figuremode {
```

```

< 6 >4 < 7\+ >8 < 6+ [_!] >
< 6 >4 <6 5 [3+] >
< _ >4 < 6 5/>4
}
>>

```



Die Unterstützung besteht aus zwei Teilen: Es gibt einen Eingabe-Modus, aktiviert durch den Befehl `\figuremode`, in dem Ziffern für den Bass als Nummern eingegeben werden können, und einen Kontext **Abschnitt “FiguredBass”** in *Programmreferenz*, der dafür sorgt, dass die entsprechenden **Abschnitt “BassFigure”** in *Programmreferenz*-Objekte auch erstellt werden.

Im Eingabemodus wird eine Gruppe von Bassziffern mit den Zeichen < and > begrenzt. Die Dauer wird nach dem >-Zeichen eingegeben.

```
<4 6>
```

**4**  
**6**

Versetzungszeichen werden durch Anhängen von -, ! und + hinter die Nummern erreicht. Ein Pluszeichen wird mit dem Befehl `\+` gesetzt, verminderte Quinten und Septimen mit 5/ und 7/.

```
<4- 6+ 7!> <5++> <3--> <7/> r <6\+ 5/>
```

**b4 x5 b3 7**      **+6**  
**#6**      **5**  
**b7**

Pausen können mit `_` notiert werden. Klammern werden mit `[` und `]` gesetzt. Text kann auch eingefügt werden nach den Regeln im **Abschnitt B.6 [Text markup commands]**, Seite 135.

```
< [4 6] 8 [_! 12] > < 5 \markup { \number 6 \super (1) } >
```

**[4] 5**  
**[6] 6** <sup>(1)</sup>  
**8**  
**[b]**  
**12**

Es ist auch möglich, Vortsetzungslinien für wiederholte Ziffern zu benutzen.

```

<<
\new Staff {
  \clef bass
  c4 c c
}
\figures {

```

```
\set useBassFigureExtenders = ##t
<4 6> <3 6> <3 7>
}
```



4 3—  
6— 7

In diesem Fall werden wiederholte Ziffern immer durch eine Linie ersetzt.

Der `FiguredBass`-Kontext kümmert sich nicht um die aktuelle Basslinie. Als Konsequenz müssen eventuell zusätzliche Ziffern eingefügt werden, um Linien unter alle Noten zu erhalten, und die Verwendung von `\!` kann nötig werden, um zu lange Linien zu stoppen.



6 ————— 5      6    6  
b4 ————— b5

Bei der Benutzung von Linien werden gemeinsame Ziffern immer in die gleiche vertikale Position gesetzt. Wenn dies nicht gewünscht ist, können Pausen (`r`) eingefügt werden. Dadurch wird die vorangegangene Positionierung aufgehoben. So kann man etwa

```
<4 6>8 r8
```

anstelle von

```
<4 6>4
```

schreiben.

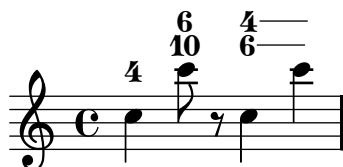
Versetzungszeichen und Pluszeichen können vor oder nach den Ziffern stehen, abhängig von den `figuredBassAlterationDirection`- und `figuredBassPlusDirection`-Eigenschaften.

+6 #5 6      +6 5# 6      6+ 5# 6      6+ #5 6  
b4            4b            4b            b4

Auch wenn die Unterstützung für bezifferten Bass anscheinend an die Akkord-Notation erinnert, ist sie doch sehr viel einfacher. Der Modus `\figuremode` speichert einfach die Ziffern und der Abschnitt [“FiguredBass”](#) in *Programmreferenz*-Kontext gibt sie so aus, wie sie notiert werden. Es gibt also keine Umrechnung von Tonhöhen und die Notation wird in der MIDI-Datei nicht umgesetzt.

Programintern erstellt die Funktion einfache Textbeschriftung. Somit sind alle Textbeschriftungsbefehle verfügbar, um die Formatierungen zu verändern. So kann zum Beispiel die vertikale Position der Ziffern mit `baseline-skip` gesetzt werden.

Bassziffern können auch direkt einem Notensystemkontext (`Staff`) hinzugefügt werden. In diesem Fall wird ihre vertikale Position automatisch bestimmt.



## Übliche Veränderungen der Einstellungen

In der Standardeinstellung werden die Bassziffern über den Noten ausgegeben. Um Ziffern unter den Noten zu erhalten, kann der Befehl

```
\override Staff.BassFigureAlignmentPositioning #'direction = #DOWN
```

benutzt werden.

## Bekannte Probleme und Warnungen

Wenn Bassziffern über dem Notensystem mit Ergänzungslinien und `implicitBassFigures` benutzt werden, kann es vorkommen, dass die Linien vertauscht werden. Besonders kritisch wird es, wenn mehrere Bassziffern sich überschneidende Linien haben. Um dieses Problem zu vermeiden, sollte `stacking-dir` im `BassFigureAlignment` benutzt werden.

## Siehe auch

Programmreferenz: Abschnitt “`BassFigure`” in *Programmreferenz*-, Abschnitt “`BassFigureAlignment`” in *Programmreferenz*-, Abschnitt “`BassFigureLine`” in *Programmreferenz*-, Abschnitt “`BassFigureBracket`” in *Programmreferenz*- und Abschnitt “`BassFigureContinuation`” in *Programmreferenz*-Objekte und der Abschnitt “`FiguredBass`” in *Programmreferenz*-Kontext.

## Eingabe des Generalbass’

### Generalbass anzeigen

## 2.8 Notation von alter Musik

### 2.8.1 Einführung in die Notation Alter Musik

#### 2.8.1.1 Notation von Alter Musik, die unterstützt wird

### 2.8.2 Alternative Notenzeichen

#### 2.8.2.1 Notenköpfe Alter Musik

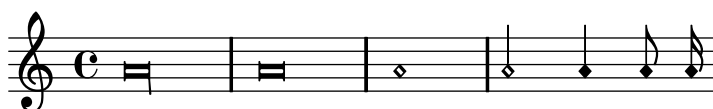
Für die Notation Alter Musik kann ein Notenkopfstil ausgewählt werden, der sich vom Standard (`default`) unterscheidet. Dies wird erreicht, indem die `style`-Eigenschaft der Notenkopf-(Abschnitt “`NoteHead`” in *Programmreferenz*)-Objekte auf einen der Werte `baroque`, `neomensural`, `mensural` oder `petrucci` gesetzt wird. Der barocke (`baroque`) Stil unterscheidet sich vom Standard (`default`) nur in sofern, als eine viereckige Form für die Brevis (`\breve`) benutzt wird. Der `neomensural`-Stil unterscheidet sich vom barocken Stil darin, dass hier rhomboide Notenköpfe für ganze Noten und kleinere Notenwerte eingesetzt werden. Hälse werden über oder unter den Notenköpfen zentriert. Dieser Stil ist vor allem dann sinnvoll, wenn mensurale Musik transkribiert werden soll, etwa für ein Incipit. Der mensurale (`mensural`) Stil erstellt Notenköpfe, die das Aussehen historischer Drucke des 16. Jahrhunderts imitieren.



Der *petrucci*-Stil schließlich imitiert auch historische Drucke, verwendet allerdings größere Notenköpfe.

Das folgende Beispiel soll den neomensuralen Stil demonstrieren:

```
\set Score.skipBars = ##t
\override NoteHead #'style = #'neomensural
a'\longa a'\breve a'1 a'2 a'4 a'8 a'16
```



Für die Notation des Gregorianischen Chorals werden vom [Abschnitt “Vaticana\\_ligature\\_engraver”](#) in *Programmreferenz* automatisch die richtigen Notenköpfe ausgewählt, so dass man den Stil nicht manuell setzen muss. Trotzdem kann der Stil manuell verändert werden, z. B. auf den Wert `vaticana_punctum`, um Punctum-Neumen zu produzieren. Gleichfalls erstellt der [Abschnitt “Mensural\\_ligature\\_engraver”](#) in *Programmreferenz* automatisch Ligaturen der Mensuralnotation. Siehe [Abschnitt 2.8.3.4 \[Ligaturen\]](#), Seite 94 zu einer Übersicht über die Funktion und Notation von Ligaturen.

## Siehe auch

[Abschnitt B.5 \[Notenkopfstile\]](#), Seite 134 stellt einen Überblick über alle verfügbaren Notenkopfstile zur Verfügung.

### 2.8.2.2 Versetzungszeichen Alter Musik

Mit der `glyph-name-alist`-Eigenschaft der Versetzungszeichen- ([Abschnitt “Accidental”](#) in *Programmreferenz*)- und Vorzeichen- ([Abschnitt “KeySignature”](#) in *Programmreferenz*)-Eigenschaften können Vorzeichen und Versetzungszeichen für die Alte Musik ausgewählt werden.

**vaticana medicaea hufnagel mensural**



Wie zu sehen ist, werden nicht alle Versetzungszeichen von jedem Stil unterstützt. Wenn versucht wird, ein Versetzungszeichen zu notieren, das von einem bestimmten Stil nicht unterstützt wird, wechselt LilyPond zu einem anderen Stil, wie in dem Beispiel `ancient-accidentals.ly` demonstriert wird.

Ähnlich wie Versetzungszeichen können auch die Vorzeichen für die Angabe der Tonart verändert werden, indem die `glyph-name-alist`-Eigenschaft des [Abschnitt “KeySignature”](#) in *Programmreferenz*-Objektes gesetzt wird.

## Siehe auch

In diesem Handbuch: [Abschnitt 1.1 \[Tonhöhen\]](#), Seite 1, [\[Versetzungszeichen\]](#), Seite 3 und [\[Automatische Versetzungszeichen\]](#), Seite 10 geben eine allgemeine Einführung in die Benutzung von Versetzungszeichen. Der [Abschnitt \[Tonartbezeichnung\]](#), Seite 9 zeigt die allgemeine Benutzung von Vorzeichen.

Programmreferenz: [Abschnitt “KeySignature”](#) in *Programmreferenz*.

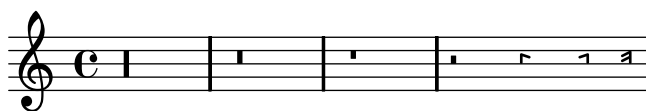
Beispiele: [Abschnitt “Ancient notation”](#) in *Beispiele*.

### 2.8.2.3 Pausen Alter Musik

Besondere Pausensymbole für die Notation der Alten Musik können mit der `style`-Eigenschaft des graphischen Objektes (grob) „Pause“ ([Abschnitt “Rest” in Programmreferenz](#)) angewählt werden. Unterstützte Stile sind klassisch (`classical`), `neomensural` und `mensural`. Der klassische (`classical`) Stil unterscheidet sich vom Standardstil (`default`) nur darin, dass die Viertelpause wie eine gespiegelte Achtelpause aussieht. Der neomensurale Stil eignet sich gut, um z. B. das Incipit von transkribierter Musik zu notieren. Der mensurale Stil ahmt die Form von Pausen nach, wie man sie in Drucken des 16. Jahrhunderts finden kann.

Das nächste Beispiel demonstriert den neomensuralen (`neomensural`) Stil:

```
\set Score.skipBars = ##t
\override Rest #'style = #'neomensural
r\longa r\breve r1 r2 r4 r8 r16
```



Es gibt keine 32-stel- und 64-stel-Pausen für den mensuralen oder neomensuralen Stil. Anstatt dessen werden die Pausenformen des Standardstiles verwendet. Vgl. eine Liste aller vorhandenen Pausen in `pitches, rests`.

Für die Notation des Gregorianischen Chorals gibt es keine Pausen; anstelle dessen werden [Abschnitt 2.8.3.3 \[Divisiones\], Seite 94](#) verwendet.


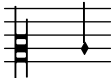
### Siehe auch

In diesem Handbuch: Der Abschnitt [Abschnitt 1.2.2.1 \[Pausen\], Seite 17](#) enthält eine allgemeine Einführung zur Benutzung von Pausen.

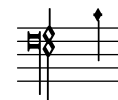
### 2.8.2.4 Schlüssel Alter Musik

LilyPond unterstützt eine große Anzahl von Notenschlüsseln, von denen eine ganze Anzahl für die Alte Musik geeignet ist.

In der Tabelle unten werden alle Schlüssel für die Alte Musik gezeigt, die mit dem `\clef`-Befehl erreicht werden. Manche Schlüssel benutzen den selben Schlüssel, unterscheiden sich aber in der Notenlinie, auf der der Schlüssel notiert wird. In diesem Fällen ist eine Nummer im Schlüsselnamen eingefügt. Man kann aber trotzdem eine beliebige Nummer erzwingen, wie es im Abschnitt [\[Notenschlüssel\], Seite 8](#) beschrieben wird. Die Note, die rechts von jedem Schlüssel gesetzt ist, zeigt das `c'` in Bezug zu dem jeweiligen Schlüssel.

Beschreibung	Unterstützte Schlüssel	Beispiel
Mensuraler C-Schlüssel im modernen Stil	<code>neomensural-c1</code> , <code>neomensural-c2</code> , <code>neomensural-c3</code> , <code>neomensural-c4</code>	
Mensuraler C-Schlüssel im Petrucci-Stil, zur Benutzung auf verschiedenen Notenlinien (im Beispiel den Schlüssel auf der zweiten Linie)	<code>petrucci-c1</code> , <code>petrucci-c2</code> , <code>petrucci-c3</code> , <code>petrucci-c4</code> , <code>petrucci-c5</code>	

Mensuraler F-Schlüssel im Petrucci-Stil      `petrucci-f`



Mensuraler G-Schlüssel im Petrucci-Stil      `petrucci-g`



Mensuraler C-Schlüssel im historischen Stil      `mensural-c1, mensural-c2, mensural-c3, mensural-c4`



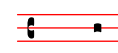
Mensuraler F-Schlüssel im historischen Stil      `mensural-f`



Mensuraler G-Schlüssel im historischen Stil      `mensural-g`



Do-Schlüssel der Editio Vaticana      `vaticana-do1, vaticana-do2, vaticana-do3`



Fa-Schlüssel der Editio Vaticana      `vaticana-fa1, vaticana-fa2`



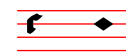
Do-Schlüssel der Editio Medicaea      `medicaea-do1, medicaea-do2, medicaea-do3`



Fa-Schlüssel der Editio Medicaea      `medicaea-fa1, medicaea-fa2`



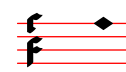
Hufnagel Do-Schlüssel für den historischen Stil      hufnagel-do1, hufnagel-do2, hufnagel-do3



Hufnagel Fa-Schlüssel für den historischen Stil      hufnagel-fa1, hufnagel-fa2



Kombinierter Hufnagelschlüssel für den historischen Stil      Do/Fa- hufnagel-do-fa



*Moderner Stil* bedeutet: „Wie in modernen Editionen von transkribierter Mensuralmusik benutzt.“

*Petrucchi-Stil* bedeutet: „Inspiriert von Drucken, die der berühmte Notensetzer Petrucci (1466–1539) produziert hat.“

*Historischer Stil* bedeutet: „Wie in anderen als Petruccis Editionen gedruckt oder geschrieben wurde.“

*Editio XXX-Stil* bedeutet: „Wie in der Editio XXX gedruckt wird.“

Petrucchi verwendete C-Schlüssel mit unterschiedlich balanciertem vertikalem Balken auf der linken Seite in Abhängigkeit davon, auf welcher Notenlinie der Schlüssel gesetzt wird.

## Siehe auch

In diesem Handbuch: siehe [\[Notenschlüssel\]](#), Seite 8.

## Bekannte Probleme und Warnungen

Der mensurale G-Schlüssel ist als Petrucci-G-Schlüssel deklariert.

### 2.8.2.5 Fähnchen Alter Musik

Mit der Fähnchen-(`flag-style`)-Eigenschaft der graphischen Objekte „Hals“ ([Abschnitt “Stem” in Programmreferenz](#)) können auch Fähnchen passend zu den Notenköpfen der Alten Musik gesetzt werden. Neben dem Standardstil (`default`) ist auch ein mensuraler Stil (`mensural`) unterstützt.

```
\override Stem #'flag-style = #'mensural
\override Stem #'thickness = #1.0
\override NoteHead #'style = #'mensural
\autoBeamOff
c'8 d'8 e'8 f'8 c'16 d'16 e'16 f'16 c'32 d'32 e'32 f'32 s8
c''8 d''8 e''8 f''8 c''16 d''16 e''16 f''16 c''32 d''32 e''32 f''32
```





Dabei ist die innerste Fahne immer vertikal auf eine Notenlinie ausgerichtet.

Es gibt keinen eigenen Stil für die „neomensurale“ Notation. Insofern sollte für das Incipit bei der Transkription mensuraler Musik der Standardstil benutzt werden. Für die Notation des Gregorianischen Chorals gibt es keine Fähnchen.

## Bekannte Probleme und Warnungen

Die Positionierung der Fähnchen an den Halsen ist leicht verschoben seit einer Änderung in einer frühen 2.3.x-Version.

Vertikale Ausrichtung der Fähnchen an einer Notenlinie geht von der Annahme aus, dass der Hals entweder genau auf einer Notenlinie oder genau zwischen zwei Notenlinien endet. Das ist aber nicht unbedingt immer der Fall, weil LilyPond komplizierte Methoden zur Ermittlung des besten Layouts verwendet. Diese Methoden sollten aber eigentlich nicht zur Notation von mensuraler Musik eingesetzt werden.

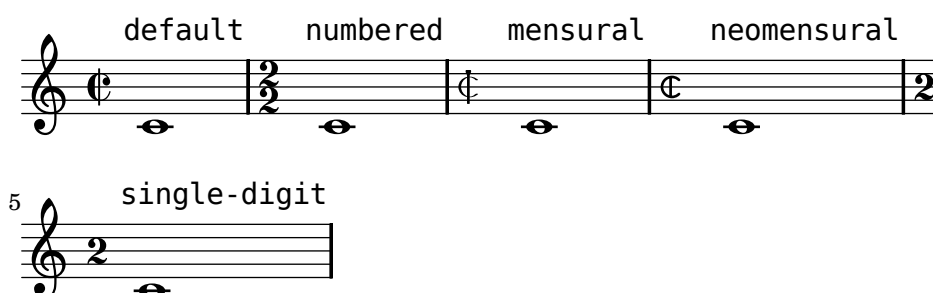
### 2.8.2.6 Taktangaben Alter Musik

LilyPond besitzt grundlegende Unterstützung für mensurale Taktangaben. Die Symbole sind starr verknüpft mit bestimmten Brüchen. Darum müssen die Werte `n` und `m` der folgenden Tabelle in den Befehl `\time n/m` eingesetzt werden, um die entsprechenden Symbole zu erhalten.

$\text{C}$	$\text{C}$	$\text{C}$	$\text{C}$
<code>\time 4/4</code>	<code>\time 6/4</code>	<code>\time 2/2</code>	<code>\time 6/8</code>
$\bigcirc$	$\textcircled{\circ}$	$\textcircled{\circ}$	$\textcircled{\circ}$
<code>\time 3/2</code>	<code>\time 3/4</code>	<code>\time 9/4</code>	<code>\time 9/8</code>
$\bigcirc$	$\textcircled{\circ}$		
<code>\time 4/8</code>	<code>\time 2/4</code>		

Mit der `style`-Eigenschaft des Objektes **Abschnitt “TimeSignature” in Programmreferenz** können die Taktarten angewählt werden. Unterstützte Stile sind: `neomensural` und `mensural`. In der Tabelle oben wurde der neomensurale Stil verwendet. Dieser Stil ist geeignet, um im Incipit von transkribierter Mensuralmusik eingesetzt zu werden. Der mensurale Stil dagegen ahmt die Form historischer Druck des 16. Jahrhunderts nach.

Im folgenden Beispiel sind die unterschiedlichen Stile dargestellt.



## Siehe auch

In diesem Handbuch: [Abschnitt 1.2.3.1 \[Taktangabe\]](#), [Seite 18](#) bietet eine allgemeine Übersicht über den Einsatz von Taktangaben.

## Bekannte Probleme und Warnungen

Die Verhältnisse der Notenwerte ändern sich nicht, wenn die Taktart (Mensur) gewechselt wird. Zum Beispiel muss das Verhältnis 1 brevis = 3 semibrevis (tempus perfectum) manuell erstellt werden, indem folgende Variable erstellt wird:

```
breveTP = #(ly:make-duration -1 0 3 2)
...
{ c \breveTP f1 }
```

Hiermit wird die Variable `breveTP` auf den Wert „3/2 mal 2 = 3 mal eine Ganze“ gesetzt.

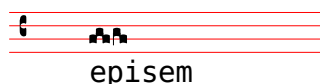
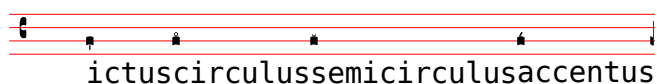
Das `old6/8alt`-Symbol (ein alternatives Symbol für 6/8) kann nicht mit dem Befehl `\time` angesprochen werden. Verwenden Sie anstatt dessen eine Textbeschriftung (`\markup`).

## 2.8.3 Zusätzliche Notenzeichen

### 2.8.3.1 Artikulationszeichen Alter Musik

Zusätzlich zu den Standardartikulationszeichen, wie sie im [Abschnitt \[Artikulationszeichen und Verzierungen\]](#), [Seite 29](#) beschrieben werden, werden auch Artikulationszeichen für die Alte Musik zur Verfügung gestellt. Diese sind darauf hin geformt, dass sie mit der Notation des Editio Vaticana-Stils verwendet werden können.

```
\include "gregorian-init.ly"
\score {
  \new VaticanaVoice {
    \override TextScript #'font-family = #'typewriter
    \override TextScript #'font-shape = #'upright
    \override Script #'padding = #-0.1
    a\ictus_"ictus" \break
    a\circulus_"circulus" \break
    a\semicirculus_"semicirculus" \break
    a\accentus_"accentus" \break
    \[ a_"episem" \episemInitium \pes b \flexa a b \episemFinis \flexa a \]
  }
}
```



## Bekannte Probleme und Warnungen

Einige Artikulationszeichen sind vertikal zu dicht an den entsprechenden Notenköpfen gesetzt.

Die Episem-Linie wird in vielen Fällen nicht angezeigt. Wenn sie angezeigt wird, ist das rechte Ende der Episem-Linie oft zu weit rechts.

### 2.8.3.2 Custodes

Ein *custos* (Plural: *custodes*; Lateinisch: „Beschützer“) ist ein Symbol, das am Ende jedes Notensystems erscheint. Es nimmt die Tonhöhe der ersten Note der nächsten Zeile vorweg und hilft damit dem Vortragenden, die Zeilenwechsel während der Vorführung zu bewältigen.

Custodes wurden bis zum 17. Jahrhundert sehr häufig in der Musiknotation eingesetzt. Heute finden sie sich nur noch in einigen bestimmten Notationsformen, etwa modernen Editionen des Gregorianischen Chorals wie die *editio vaticana*. LilyPond stellt unterschiedliche Custos-Symbole für die unterschiedlichen Notationsstile zur Verfügung.

Damit Custodes angezeigt werden, muss ein Abschnitt `“Custos_engraver”` in *Programmreferenz* im Abschnitt `“Staff”` in *Programmreferenz*-Kontext gefordert werden. Der Aufruf folgt im Rahmen des Layout-Kontextes, wie das folgende Beispiel zeigt.

```
\layout {
  \context {
    \Staff
    \consists Custos_engraver
    Custos \override #'style = #'mensural
  }
}
```

Das Ergebnis sieht ungefähr folgendermaßen aus:



Das Custos-Zeichen wird von der `style`-Eigenschaft ausgewählt. Die unterstützten Stile sind: *vaticana*, *medicaea*, *hufnagel* und *mensural*. Sie werden im folgenden Fragment demonstriert.

*vaticana* *medicaea* *hufnagel* *mensural*

## Siehe auch

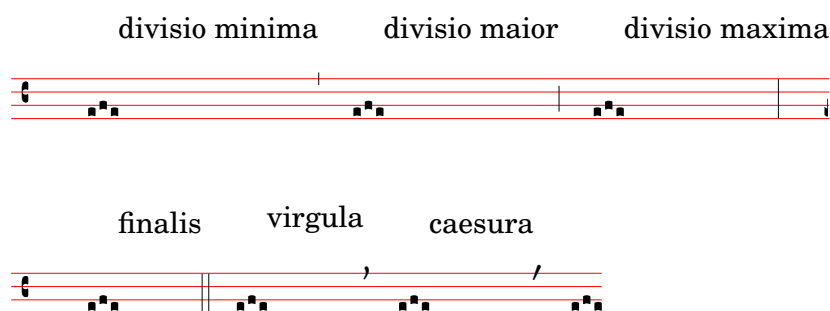
Programmreferenz: Abschnitt `“Custos”` in *Programmreferenz*.

Beispiele: Abschnitt `“Ancient notation”` in *Beispiele*.

### 2.8.3.3 Divisiones

Eine *divisio* (Plural: *divisiones*; Latein: „Teilung“) ist ein Symbol des Notensystemkontextes, das benutzt wird, um Phrasierung und Abschnitte im Gregorianischen Choral anzuzeigen. Die musikalische Bedeutung von *divisio minima*, *divisio maior* und *divisio maxima* kann beschrieben werden als kurze, mittlere und lange Pause, ungefähr wie die Atemzeichen aus dem Abschnitt [\[Atemzeichen\]](#), Seite 36. Das *finalis*-Zeichen bezeichnet nicht nur das Ende eines Chorals, sondern wird auch oft innerhalb eines Antiphons/Responsorius benutzt, um das Ende eines Abschnitts anzuzeigen.

Divisiones können benutzt werden, indem die Datei ‘gregorian-init.ly’ in die Quelldatei eingefügt wird. Hier sind die entsprechenden Definitionen schon abgelegt, so dass es genügt, die Befehle `\divisioMinima`, `\divisioMaior`, `\divisioMaxima` und `\finalis` an den entsprechenden Stellen zu schreiben. Einige Editionen verwenden eine *virgula* oder *caesura* anstelle der *divisio minima*. Darum findet sich in der Datei ‘gregorian-init.ly’ auch eine Definition für `\virgula` und `\caesura`.



### Vordefinierte Befehle

`\virgula`, `\caesura`, `\divisioMinima`, `\divisioMaior`, `\divisioMaxima`, `\finalis`.

### Siehe auch

In diesem Handbuch: [\[Atemzeichen\]](#), Seite 36.

Programmreferenz: [Abschnitt “BreathingSign”](#) in *Programmreferenz*.

Beispiele: [Abschnitt “Winds”](#) in *Beispiele*.

### 2.8.3.4 Ligaturen

Eine Ligatur ist ein graphisches Symbol das wenigstens zwei unterschiedliche Noten darstellt. Ligaturen treten ursprünglich in Manuskripten des Gregorianischen Chorals auf, um auf- oder absteigende Notensequenzen zu notieren.

Ligaturen werden eingegeben, indem die dazugehörigen Noten zwischen `\[` und `\]` eingeschlossen werden. Einige Ligaturstile benötigen zusätzliche Syntax für eine bestimmte Ligatur. In der Standardeinstellung setzt der [Abschnitt “LigatureBracket”](#) in *Programmreferenz* ganz einfach eckige Klammern über die Noten der Ligatur.

```
\transpose c c' {
  \[ g c a f d' \]
  a g f
  \[ e f a g \]
}
```





Um einen gestimmten Ligaturstil auszuwählen, muss ein entsprechender Ligatur-Engraver zum Stimmkontext hinzugefügt werden, wie in den folgenden Abschnitten erklärt wird. Nur weiße Mensuralligaturen sind unterstützt – mit Einschränkungen.

## Bekannte Probleme und Warnungen

Ligaturen benötigen von klassischer Notation unterschiedliche Platzaufteilung, was sie aber noch nicht können. Darum ist fast immer zu viel Platz zwischen Ligaturen und Zeilenumbrüche sind ungenügend. Text lässt sich auch nicht richtig an Ligaturen ausrichten.

Akzidentien dürfen nicht innerhalb von einer Ligatur gedruckt werden, sondern müssen gesammelt und vor der Ligatur ausgegeben werden.

Die Syntax verwendet immer noch den verworfenen Infix-Stil (`\[ musik. Ausdr. \]`). Für die Konsistenz soll dies geändert werden in den Postfix-Stil (`Note\[ ... Note\]`). Alternativ kann die Datei `'gregorian-init.ly'` eingefügt werden, die eine Scheme-Funktion

```
\ligature musik. Ausdr.
```

mit der selben Wirkung zur Verfügung stellt und stabil zu sein scheint.

### 2.8.3.5 Weiße Mensuralligaturen

Begrenzte Unterstützung für Ligaturen der weißen Mensuralnotation.

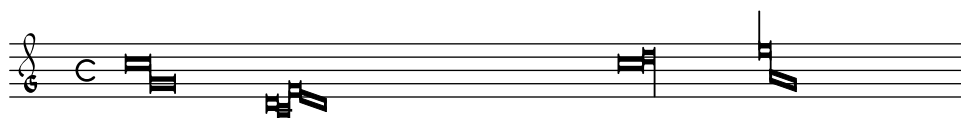
Um weiße Mensuralligaturen zu benutzen, muss innerhalb des Layout-Blocks im Abschnitt *“Voice”* in *Programmreferenz*-Kontext der Abschnitt *“Mensural\_ligature\_engraver”* in *Programmreferenz* aktiviert werden und gleichzeitig der Abschnitt *“Ligature\_bracket\_engraver”* in *Programmreferenz* (der die Klammern über den Noten setzt) entfernt werden, wie im Beispiel.

```
\layout {
  \context {
    \Voice
    \remove Ligature_bracket_engraver
    \consists Mensural_ligature_engraver
  }
}
```

Zusätzlich zu diesen Einstellungen gibt es keine eigenen Befehle, die die Form einer Ligatur bestimmen. Die Form wird vielmehr aus Tonhöhen und Tondauern der in Klammern gesetzten Noten geschlossen. Diese Herangehensweise erfordert einige Eingewöhnung, hat aber den großen Vorteil, dass der musikalische Inhalt der Ligatur dem Programm bekannt ist. Das ist nicht nur notwendig für korrekte MIDI-Ausgabe, sondern erlaubt es auch, automatische Transkriptionen von Ligaturen anzufertigen.

Eine Datei kann zum Beispiel so aussehen:

```
\set Score.timing = ##f
\set Score.defaultBarType = "empty"
\override NoteHead #'style = #'neomensural
\override Staff.TimeSignature #'style = #'neomensural
\clef "petrucci-g"
\[ c'\maxima g \]
\[ d\longa c\breve f e d \]
\[ c'\maxima d'\longa \]
\[ e'1 a g\breve \]
```



Wenn der Abschnitt “*Ligature\_bracket\_engraver*” in *Programmreferenz* nicht durch den Abschnitt “*Mensural\_ligature\_engraver*” in *Programmreferenz* ersetzt wird, werden die Noten wie folgt ausgegeben:



## Bekannte Probleme und Warnungen

Die horizontale Positionierung ist sehr schlecht.

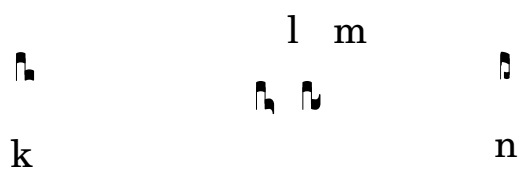
### 2.8.3.6 Ligaturen der gregorianischen Quadratnotation

Beschränkte Unterstützung für gregorianische Quadratneumen-Ligaturen (nach dem Stil der Editio Vaticana) ist vorhanden. Die wichtigsten Ligaturen können schon gesetzt werden, aber wichtige Eigenschaften anspruchsvoller Typographie wie horizontale Ausrichtung von mehreren Ligaturen, korrekte Silbenpositionierung und richtiger Umgang mit Versetzungszeichen fehlen noch.

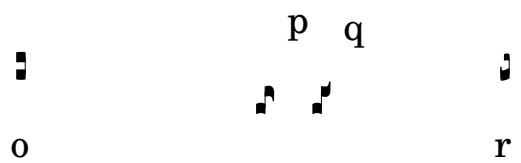
Die folgende Tabelle enthält die erweiterte Neumenliste des zweiten Bands des Antiphonale Romanum (*Liber Hymnarius*), 1983 von den Mönchen von Solesmes herausgegeben.

Neuma aut Neumarum Elementa	Figurae Rectae	Figurae Liquescentes Auctae	Figurae Liquescentes Deminutae
1. Punctum	a b ■ ◆	c d e ■ ◆ ◆	f ◆
2. Virga	g ■		
3. Apostropha vel Strophæ	h ◆	i ◆	
4. Oriscus	j ■		

5. Clivis vel Flexa



6. Podatus vel Pes



7. Pes Quassus



8. Quilisma Pes



9. Podatus Initio Debilis



10. Torculus



y



z



A

11. Torculus Initio Debilis



B



C



D

12. Porrectus



E



F



G

13. Climacus



H



I



J

14. Scandicus



K

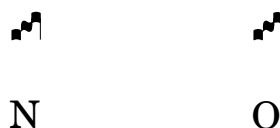


L



M

## 15. Salicus



## 16. Trigonus



Anders als in den meisten Neumennotationssystemen zeigt der Quellcode von LilyPond nicht das typographische Aussehen der Ligatur an, sondern deren musikalischen Inhalt. Der Code `\[ a \pes b \flexa g \]` etwa ergibt einen Torculus, der aus drei Punctum-Köpfen besteht, während `\[ a \flexa g \pes b \]` einen Porrectus mit einer gekrümmten Flexa und einem einzelnen Punctum ausgibt. Es gibt also keinen Befehl, der explizit eine gekrümmte Flexa setzen würde; die Entscheidung, wann diese gesetzt werden soll hängt vielmehr davon ab, welcher musikalische Inhalt dargestellt werden soll. Der Sinn dieser Herangehensweise ist es, den Inhalt von der graphischen Repräsentation zu trennen. Auf diese Art kann mit der gleichen Eingabe ein anderer gregorianischer Stil gesetzt werden, ohne die Notation zu verändern.

Die folgende Tabelle zeigt Code-Fragmente, mit denen die Ligaturen der vorigen Tabelle erstellt werden können. Der Buchstabe in der ersten Spalte jeder Zeile der unteren Tabelle zeigt an, auf welche Ligatur in der vorigen Tabelle sie sich bezieht. In der zweiten Spalte erscheint die Bezeichnung der Ligatur. Die dritte Spalte enthält das Fragment, mit dem die Ligatur erzeugt wurde, wobei `g`, `a` und `b` als Beispieltonhöhen eingesetzt werden.

#	Name	Input Language
a	Punctum	<code>\[ b \]</code>
b	Punctum Inclinatorum	<code>\[ \inclinatorum b \]</code>
c	Punctum Auctum Ascendens	<code>\[ \auctum \ascendens b \]</code>
d	Punctum Auctum Descendens	<code>\[ \auctum \descendens b \]</code>
e	Punctum Inclinatorum Auctum	<code>\[ \inclinatorum \auctum b \]</code>
f	Punctum Inclinatorum Parvum	<code>\[ \inclinatorum \deminutum b \]</code>
g	Virga	<code>\[ \virga b \]</code>
h	Strophä	<code>\[ \strophä b \]</code>

i	Stropha Aucta	<code>\[ \stropha \auctum b \]</code>
j	Oriscus	<code>\[ \oriscus b \]</code>
k	Clivis vel Flexa	<code>\[ b \flexa g \]</code>
l	Clivis Aucta Descendens	<code>\[ b \flexa \auctum \descendens g \]</code>
m	Clivis Aucta Ascendens	<code>\[ b \flexa \auctum \ascendens g \]</code>
n	Cephalicus	<code>\[ b \flexa \deminutum g \]</code>
o	Podatus vel Pes	<code>\[ g \pes b \]</code>
p	Pes Auctus Descendens	<code>\[ g \pes \auctum \descendens b \]</code>
q	Pes Auctus Ascendens	<code>\[ g \pes \auctum \ascendens b \]</code>
r	Epiphonus	<code>\[ g \pes \deminutum b \]</code>
s	Pes Quassus	<code>\[ \oriscus g \pes \virga b \]</code>
t	Pes Quassus Auctus Descendens	<code>\[ \oriscus g \pes \auctum \descendens b \]</code>
u	Quilisma Pes	<code>\[ \quilisma g \pes b \]</code>
v	Quilisma Pes Auctus Descendens	<code>\[ \quilisma g \pes \auctum \descendens b \]</code>
w	Pes Initio Debilis	<code>\[ \deminutum g \pes b \]</code>
x	Pes Auctus Descendens Initio Debilis	<code>\[ \deminutum g \pes \auctum \descendens b \]</code>
y	Torculus	<code>\[ a \pes b \flexa g \]</code>
z	Torculus Auctus Descendens	<code>\[ a \pes b \flexa \auctum \descendens g \]</code>
A	Torculus Deminutus	<code>\[ a \pes b \flexa \deminutum g \]</code>
B	Torculus Initio Debilis	<code>\[ \deminutum a \pes b \flexa g \]</code>
C	Torculus Auctus Descendens Initio Debilis	<code>\[ \deminutum a \pes b \flexa \auctum \descendens g \]</code>
D	Torculus Deminutus Initio Debilis	<code>\[ \deminutum a \pes b \flexa \deminutum g \]</code>
E	Porrectus	<code>\[ a \flexa g \pes b \]</code>
F	Porrectus Auctus Descendens	<code>\[ a \flexa g \pes \auctum \descendens b \]</code>
G	Porrectus Deminutus	<code>\[ a \flexa g \pes \deminutum b \]</code>

H	Climacus	<code>\[ \virga b \inclinatum a \inclinatum g \]</code>
I	Climacus Auctus	<code>\[ \virga b \inclinatum a \inclinatum \auctum g \]</code>
J	Climacus Deminutus	<code>\[ \virga b \inclinatum a \inclinatum \deminutum g \]</code>
K	Scandicus	<code>\[ g \pes a \virga b \]</code>
L	Scandicus Auctus Descendens	<code>\[ g \pes a \pes \auctum \descendens b \]</code>
M	Scandicus Deminutus	<code>\[ g \pes a \pes \deminutum b \]</code>
N	Salicus	<code>\[ g \oriscus a \pes \virga b \]</code>
O	Salicus Auctus Descendens	<code>\[ g \oriscus a \pes \auctum \descendens b \]</code>
P	Trigonus	<code>\[ \stroph a b \stroph a b \stroph a \]</code>

Die Ligaturen dieser Liste dienen als begrenzter, aber doch repräsentativer Vorrat an Ligaturbeispielen des Gregorianischen Chorals. Innerhalb der Ligaturbegrenzungen `\[` und `\]` kann jedoch problemlos jede nur mögliche Anzahl an Noten gesetzt werden, und Präfixe wie `\pes`, `\flexa`, `\virga`, `\inclinatum`, usw können nach Belieben gemischt werden. Die Regeln, die der Konstruktion der Ligaturen in den Tabellen zugrunde liegen, werden entsprechend angepasst. Auf diese Weise können unendlich viele Ligaturen gesetzt werden.

Augmentum-Punkte, auch *morae* genannt, werden mit dem Befehl `\augmentum` hinzugefügt. `\augmentum` ist allerdings als eigene musikalische Funktion gebaut und nicht als ein Notenpräfix. Insofern hat der Befehl in diesem Kontext: `\augmentum \virga c` keine sichtbaren Auswirkungen. Erst mit `\virga \augmentum c` oder `\augmentum {\virga c}` funktionieren beide Befehle. Es ist auch möglich, mit `\augmentum {a g}` die Schreibweise `\augmentum a \augmentum g` abzukürzen.

```
\include "gregorian-init.ly"
\score {
  \new VaticanaVoice {
    \[ \augmentum a \flexa \augmentum g \]
    \augmentum g
  }
}
```



## Vordefinierte Befehle

Folgende Notenpräfixe sind unterstützt:

`\virga`, `\stroph a`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.

Präfixe können kombiniert werden, wenn es hier auch Begrenzungen gibt. Zum Beispiel können die Präfixe `\descendens` oder `\ascendens` vor einer Note geschrieben werden, aber nicht beide für die selbe Note. Zwei benachbarte Noten können mit den `\pes` und `\flexa`-Infixen verbunden werden, um eine steigende bzw. fallende Melodielinie zu notieren.

Die musikalische Funktion `\augmentum` muss benutzt werden, um augmentum-Punkte hinzuzufügen.

## Bekannte Probleme und Warnungen

Wenn ein `\augmentum`-Punkt am Ende des letzten Systems innerhalb einer Ligatur gesetzt wird, ist er vertikal etwas falsch positioniert. Als Abhilfe kann eine unsichtbare Note (z. B. `s8`) als letzte Note im System eingegeben werden.

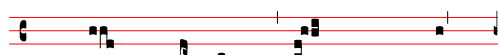
`\augmentum` sollte als Präfix implementiert sein, nicht als eigene musikalische Funktion, so dass `\augmentum` mit den anderen Präfixen in arbiträrer Reihenfolge notiert werden kann.

### 2.8.4 Vordefinierte Umgebungen

#### 2.8.4.1 Gregorianische Gesangs-Kontexte

Die vordefinierten Kontexte `VaticanaVoiceContext` (für eine gregorianische Stimme) und `VaticanaStaffContext` (für ein gregorianisches Notensystem) können eingesetzt werden, um Gregorianischen Choral im Stil der Editio Vaticana zu setzen. Diese Kontexte initialisieren alle relevanten Eigenschaften für das Notensystem und die graphischen Objekte, so dass unmittelbar mit der Notation begonnen werden kann. Siehe das folgende Beispiel:

```
\include "gregorian-init.ly"
\score {
  <<
    \new VaticanaVoice = "cantus" {
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \]
      f \divisioMinima
      \[ f\melisma \pes a c' c' \pes d'\melismaEnd \]
      c' \divisioMinima \break
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \] f \divisioMinima
    }
    \new Lyrics \lyricsto "cantus" {
      San- ctus, San- ctus, San- ctus
    }
  >>
}
```



San- ctus, San- ctus,



San- ctus

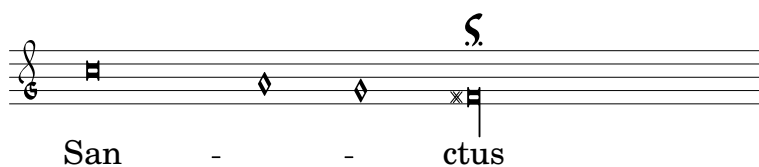
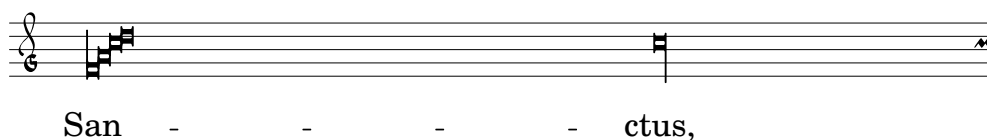
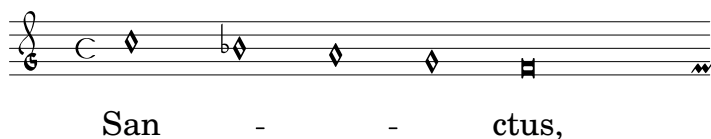
#### 2.8.4.2 Mensural-Kontexte

Die vordefinierten Kontexte `MensuralVoiceContext` und `MensuralStaffContext` können eingesetzt werden, um ein Stück in Mensuralnotations zu schreiben. Die Kontexte initialisieren alle



relevanten Eigenschaften der Kontexte und graphischen Objekte, so dass unmittelbar mit der Notation begonnen werden kann. Siehe das folgende Beispiel:

```
\score {
  <<
    \new MensuralVoice = "discantus" \transpose c c' {
      \override Score.BarNumber #'transparent = ##t {
        c'1\melisma bes a g\melismaEnd
        f\breve
        \[ f1\melisma a c'\breve d'\melismaEnd \]
        c'\longa
        c'\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}
```



## 2.8.5 Transkription Alter Musik

### 2.8.5.1 Alte und moderne Edition aus einer Quelldatei

### 2.8.5.2 Incipite

### 2.8.5.3 Mensurstriche

### 2.8.5.4 Gregorianischen Choral transkribieren

## 2.8.6 Herausgeberische Anmerkungen

### 2.8.6.1 Vorgeschlagene Versetzungszeichen

In der europäischen Musik vor 1600 wurden vom Sänger oftmals chromatische Alterationen erwartet, die nicht notiert wurden. Diese Praxis wird „Musica Ficta“ genannt. In modernen Transkription werden die Versetzungszeichen üblicherweise oberhalb der Noten gesetzt.

Unterstützung für solche empfohlenen Versetzungszeichen ist implementiert und kann aktiviert werden, indem die Eigenschaft `suggestAccidentals` auf wahr gesetzt wird. Siehe auch das Beispiel.

```
fis gis
\set suggestAccidentals = ##t
ais bis
```



### Siehe auch

Programmreferenz: [Abschnitt “Accidental\\_engraver”](#) in *Programmreferenz*-Setzer und das [Abschnitt “AccidentalSuggestion”](#) in *Programmreferenz*-Objekt.

### 2.8.6.2 Notation barocker Rhythmen

## 2.9 Weltmusik

### 2.9.1 Arabische Musik

#### References for Arabic music

Arabic note names

Arabic key signatures

Arabic time signatures

Arabic music example

Weitere Literatur

## 3 Allgemeine Eingabe und Ausgabe

### 3.1 Eingabestruktur

#### 3.1.1 Struktur einer Partitur

#### 3.1.2 Mehrere Partituren in einem Buch

#### 3.1.3 Die Dateistruktur

### 3.2 Titel

#### 3.2.1 Titel erstellen

#### 3.2.2 Eigene Titel

#### 3.2.3 Verweis auf die Seitenzahlen

#### 3.2.4 Inhaltsverzeichnis

### 3.3 Arbeiten an Eingabe-Dateien

#### 3.3.1 LilyPond-Dateien einfügen

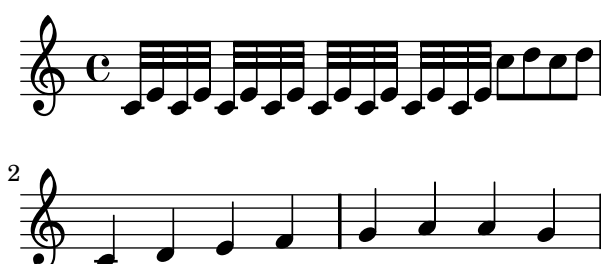
#### 3.3.2 Verschiedene Editionen aus einer Quelldatei

#### Variablen benutzen

#### Marken benutzen

Mit ein bisschen Anpassung können alle Wiederholungstypen auch in der MIDI-Datei wiedergegeben werden. Das wird erreicht durch die `\unfoldRepeats`-Funktion. Hiermit werden alle Wiederholungen, welcher Art auch immer, in notengetreue Wiederholungen umgewandelt, die Noten werden also wiederholt ausgegeben.

```
\unfoldRepeats {
  \repeat tremolo 8 {c'32 e' }
  \repeat percent 2 { c''8 d'' }
  \repeat volta 2 {c'4 d' e' f'}
  \alternative {
    { g' a' a' g' }
    {f' e' d' c' }
  }
}
\bar "|."
```





Wenn man eine Partitur schreibt, die diesen `\unfoldRepeats`-Befehl für die MIDI-Ausgabe benutzt, muss man zwei `\score`-Umgebungen schreiben: eine für die MIDI-Ausgabe, wo die Wiederholungen ausgeschrieben werden, und eine für die gedruckte Notation, in der Klammern, Tremolo und Prozent-Wiederholungen benutzt werden. Zum Beispiel:

```
\score {
  ..music..
  \layout { .. }
}
\score {
  \unfoldRepeats ..music..
  \midi { .. }
}
```

### 3.3.3 Zeichenkodierung

### 3.3.4 LilyPond-Notation anzeigen

## 3.4 Ausgabe kontrollieren

### 3.4.1 Notationsfragmente extrahieren

### 3.4.2 Korrigierte Musik überspringen

## 3.5 MIDI-Ausgabe

### 3.5.1 MIDI-Dateien erstellen

#### Instrumentenbezeichnungen

### 3.5.2 Der MIDI-Block

### 3.5.3 Was geht in die MIDI-Ausgabe

#### In MIDI unterstützt

#### In MIDI nicht unterstützt

### 3.5.4 Wiederholungen im MIDI

### 3.5.5 MIDI-Lautstärke kontrollieren

#### Dynamik-Zeichen

#### MIDI-Lautstärke

#### Verschiedene Instrumente angleichen (i)

#### Verschiedene Instrumente angleichen (ii)

## 4 Abstände

### 4.1 Papier und Seiten

#### 4.1.1 Papierformat

#### 4.1.2 Seitenformatierung

### 4.2 Notenlayout

#### 4.2.1 Die Notensystemgröße einstellen

#### 4.2.2 Partiturlayout

### 4.3 Umbrüche

#### 4.3.1 Zeilenumbrüche

#### 4.3.2 Seitenumbrüche

#### 4.3.3 Optimale Seitenumbrüche

#### 4.3.4 Optimale Umbrüche zum Blättern

#### 4.3.5 Minimale Seitenumbrüche

#### 4.3.6 Ausdrückliche Umbrüche

#### 4.3.7 Eine zusätzliche Stimme für Umbrüche benutzen

### 4.4 Vertikale Abstände

#### 4.4.1 Vertikale Abstände innerhalb eines Systems

#### 4.4.2 Vertikale Abstände zwischen Systemen

#### 4.4.3 Explizite Positionierung von Systemen

#### 4.4.4 Vertikale Abstände mit zwei Durchgängen

#### 4.4.5 Vermeidung von vertikalen Zusammenstößen

### 4.5 Horizontale Abstände

#### 4.5.1 Überblick über horizontale Abstände

#### 4.5.2 Eine neuer Bereich mit anderen Abständen

#### 4.5.3 Horizontale Abstände verändern

#### 4.5.4 Zeilenlänge

### 4.5.5 Proportionale Notation

## 4.6 Die Musik auf weniger Seiten zwingen

### 4.6.1 Abstände anzeigen lassen

### 4.6.2 Abstände verändern

Manchmal bleiben nur noch ein oder zwei Systeme auf der letzten Seite übrig. Das ist immer ärgerlich, besonders wenn es scheint, dass auf den vorigen Seiten genug Platz ist, um die Systeme noch unterzubringen.

Wenn man versucht, das Layout zu verändern, kommt einem der Befehl `annotate-spacing` zu Hilfe. Mit diesem Befehl erhält man die Werte von verschiedenen Abstandsbefehlen ausgedruckt, mehr Information im Kapitel [Abschnitt 4.6.1 \[Abstände anzeigen lassen\]](#), Seite 108. Anhand dieser Angaben kann dann entschieden werden, welche Werte verändert werden müssen.

Neben Rändern gibt es nämlich weitere Optionen, Platz zu sparen:

- LilyPond kann die Systeme so dicht wie möglich platzieren (damit so viele Systeme wie möglich auf eine Seite passen), aber sie dann so anordnen, dass kein weißer Rand unten auf der Seite entsteht.

```
\paper {
  between-system-padding = #0.1
  between-system-space = #0.1
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Die Anzahl der Systeme kann reduziert werden (wenn LilyPond die Musik auf 11 Systeme verteilt, kann man die Benutzung von nur 10 Systemen erzwingen).

```
\paper {
  system-count = #10
}
```

- Vermeidung von Objekten, die den vertikalen Abstand von Systemen vergrößern, hilft oft. Die Verwendung von Klammern bei Wiederholungen etwa braucht mehr Platz. Wenn die Noten innerhalb der Klammern auf zwei Systeme verteilt sind, brauchen sie mehr Platz, als wenn sie nur auf einer Zeile gedruckt werden.

Ein anderes Beispiel ist es, Dynamik-Zeichen, die besonders weit „hervorstehen“, zu verschieben.

```
\relative c' {
  e4 c g\ff c
  \override DynamicLineSpanner #'padding = #-1.8
  \override DynamicText #'extra-offset = #'(-2.1 . 0)
  e4 c g\ff c
}
```



- Die horizontalen Abstände können mit der `SpacingSpanner`-Eigenschaft verändert werden. Siehe [Abschnitt 4.5.3 \[Horizontale Abstände verändern\]](#), Seite 107 für Einzelheiten.

9

## 5 Standardeinstellungen verändern

### 5.1 Interpretationsumgebungen

#### 5.1.1 Was sind Umgebungen?

Score - Meister aller Kontexte

Oberste Kontexte – Container für Systeme

Mittlere Kontexte – Systeme

Unterste Kontexte – Stimmen

#### 5.1.2 Umgebungen erstellen

#### 5.1.3 Umgebungs-Plugins verändern

#### 5.1.4 Die Standardeinstellungen von Umgebungen ändern

#### 5.1.5 Neue Umgebungen definieren

#### 5.1.6 Umgebungen aneinander ausrichten

### 5.2 Die Referenz der Programminterna erklärt

#### 5.2.1 Zurechtfinden in der Programmreferenz

#### 5.2.2 Layout-Schnittstellen

#### 5.2.3 Die Grob-Eigenschaften

#### 5.2.4 Benennungsübereinkommen

### 5.3 Eigenschaften verändern

#### 5.3.1 Überblick über verändernde Eigenschaften

#### 5.3.2 Der `\set`-Befehl

#### 5.3.3 Der `\override`-Befehl

#### 5.3.4 `\set` versus `\override`

#### 5.3.5 Der `\tweak`-Befehl

### 5.4 Nützliche Konzepte und Eigenschaften

#### 5.4.1 Eingabe-Modi

#### 5.4.2 Richtung und Platzierung



### 5.4.3 Abstände und Maße

### 5.4.4 Strecker

## 5.5 Übliche Eigenschaften

### 5.5.1 Die Sichtbarkeit von Objekten kontrollieren

Einen stencil entfernen

Objekten unsichtbar machen

Objekte weiß malen

break-visibility (unsichtbar machen) benutzen

Besonderheiten

### 5.5.2 Zeilenstile

### 5.5.3 Drehen von Objekten

### 5.5.4 Umgebungen ausrichten

## 5.6 Fortgeschrittene Optimierungen

### 5.6.1 Vertikale Gruppierung der grafischen Objekte („grob“s)

### 5.6.2 Enden von Streckern verändern

### 5.6.3 stencils verändern

### 5.6.4 Formen verändern

## 5.7 Erklärung von speziellen Optimierungen

### 5.7.1 Alte Kontexte

## 6 Schnittstellen für Programmierer

### 6.1 Musikalische Funktionen

#### 6.1.1 Überblick über musikalische Funktionen

#### 6.1.2 Einfache Ersetzungsfunktionen

#### 6.1.3 Paarige Ersetzungsfunktionen

#### 6.1.4 Mathematik in Funktionen

#### 6.1.5 Leere Funktionen

#### 6.1.6 Funktionen ohne Argumente

#### 6.1.7 Überblick über vorhandene musikalische Funktionen

`acciaccatura` - *music* (music)  
(undocumented; fixme)

`addChordShape` - *key-symbol* (symbol) *shape-string* (string)  
(undocumented; fixme)

`addInstrumentDefinition` - *name* (string) *lst* (list)  
(undocumented; fixme)

`addQuote` - *name* (string) *music* (music)  
(undocumented; fixme)

`afterGrace` - *main* (music) *grace* (music)  
(undocumented; fixme)

`allowPageTurn`  
(undocumented; fixme)

`applyContext` - *proc* (procedure)  
(undocumented; fixme)

`applyMusic` - *func* (procedure) *music* (music)  
(undocumented; fixme)

`applyOutput` - *ctx* (symbol) *proc* (procedure)  
(undocumented; fixme)

`appoggiatura` - *music* (music)  
(undocumented; fixme)

`assertBeamQuant` - *l* (pair) *r* (pair)  
(undocumented; fixme)

`assertBeamSlope` - *comp* (procedure)  
(undocumented; fixme)

`autochange` - *music* (music)  
(undocumented; fixme)

`balloonGrobText` - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)  
(undocumented; fixme)

**balloonText** - *offset* (pair of numbers) *text* (markup)  
 (undocumented; fixme)

**bar** - *type* (string)  
 (undocumented; fixme)

**barNumberCheck** - *n* (integer)  
 (undocumented; fixme)

**bendAfter** - *delta* (unknown)  
 (undocumented; fixme)

**breathe** (undocumented; fixme)

**clef** - *type* (string)  
 (undocumented; fixme)

**cueDuring** - *what* (string) *dir* (direction) *main-music* (music)  
 (undocumented; fixme)

**displayLilyMusic** - *music* (music)  
 (undocumented; fixme)

**displayMusic** - *music* (music)  
 (undocumented; fixme)

**endSpanners** - *music* (music)  
 (undocumented; fixme)

**featherDurations** - *factor* (moment) *argument* (music)  
 (undocumented; fixme)

**grace** - *music* (music)  
 (undocumented; fixme)

**includePageLayoutFile**  
 (undocumented; fixme)

**instrumentSwitch** - *name* (string)  
 (undocumented; fixme)

**keepWithTag** - *tag* (symbol) *music* (music)  
 (undocumented; fixme)

**killCues** - *music* (music)  
 (undocumented; fixme)

**label** - *label* (symbol)  
 (undocumented; fixme)

**makeClusters** - *arg* (music)  
 (undocumented; fixme)

**musicMap** - *proc* (procedure) *mus* (music)  
 (undocumented; fixme)

**noPageBreak**  
 (undocumented; fixme)

**noPageTurn**  
 (undocumented; fixme)

**octaveCheck** - *pitch-note* (music)  
 (undocumented; fixme)

`oldaddyrics` - *music* (music) *lyrics* (music)  
 (undocumented; fixme)  
`ottava` - *octave* (number)  
 (undocumented; fixme)  
`overrideProperty` - *name* (string) *property* (symbol) *value* (any type)  
 (undocumented; fixme)  
`pageBreak`  
 (undocumented; fixme)  
`pageTurn` (undocumented; fixme)  
`parallelMusic` - *voice-ids* (list) *music* (music)  
 (undocumented; fixme)  
`parenthesize` - *arg* (music)  
 (undocumented; fixme)  
`partcombine` - *part1* (music) *part2* (music)  
 (undocumented; fixme)  
`pitchedTrill` - *main-note* (music) *secondary-note* (music)  
 (undocumented; fixme)  
`pointAndClickOff`  
 (undocumented; fixme)  
`pointAndClickOn`  
 (undocumented; fixme)  
`quoteDuring` - *what* (string) *main-music* (music)  
 (undocumented; fixme)  
`removeWithTag` - *tag* (symbol) *music* (music)  
 (undocumented; fixme)  
`resetRelativeOctave` - *reference-note* (music)  
 (undocumented; fixme)  
`rightHandFinger` - *finger* (number or string)  
 (undocumented; fixme)  
`scaleDurations` - *fraction* (pair of numbers) *music* (music)  
 (undocumented; fixme)  
`scoreTweak` - *name* (string)  
 (undocumented; fixme)  
`shiftDurations` - *dur* (integer) *dots* (integer) *arg* (music)  
 (undocumented; fixme)  
`spacingTweaks` - *parameters* (list)  
 (undocumented; fixme)  
`storePredefinedDiagram` - *chord* (music) *tuning* (list) *terse-definition* (string)  
 (undocumented; fixme)  
`tag` - *tag* (symbol) *arg* (music)  
 (undocumented; fixme)  
`tocItem` - *text* (markup)  
 Add a line to the table of content, using the `tocItemMarkup` paper variable markup

`transposedCueDuring` - *what* (string) *dir* (direction) *pitch-note* (music) *main-music* (music)  
(undocumented; fixme)

`transposition` - *pitch-note* (music)  
(undocumented; fixme)

`tweak` - *sym* (symbol) *val* (any type) *arg* (music)  
(undocumented; fixme)

`unfoldRepeats` - *music* (music)  
(undocumented; fixme)

`withMusicProperty` - *sym* (symbol) *val* (any type) *music* (music)  
(undocumented; fixme)

## 6.2 Schnittstelle für Programmierer

### 6.2.1 Eingabevariablen und Scheme

### 6.2.2 Interne Repräsentation der Musik

## 6.3 Komplizierte Funktionen erstellen

### 6.3.1 Musikalische Funktionen darstellen

### 6.3.2 Eigenschaften von Musikobjekten

### 6.3.3 Verdoppelung einer Note mit Bindebögen (Beispiel)

### 6.3.4 Artikulationszeichen zu Noten hinzufügen (Beispiel)

## 6.4 Programmierungsschnittstelle für Textbeschriftungen

### 6.4.1 Beschriftungskonstruktionen in Scheme

### 6.4.2 Wie Beschriftungen intern funktionieren

### 6.4.3 Neue Definitionen von Beschriftungsbefehlen

### 6.4.4 Neue Definitionen von Beschriftungsbefehlen für Listen

## 6.5 Kontexte für Programmierer

### 6.5.1 Kontextauswertung

### 6.5.2 Eine Funktion auf alle Layout-Objekte anwenden

## 6.6 Scheme-Vorgänge als Eigenschaften

## 6.7 TODO moved into scheme

### 6.7.1 Scheme-Code anstelle von `weak` verwenden

### 6.7.2 Schwierige Korrekturen

## Anhang A Literatur

Wenn Sie mehr über Notation und den Notenstich erfahren wollen, sind hier einige interessante Titel gesammelt.

### *Ignatzek 1995*

Klaus Ignatzek, Die Jazzmethode für Klavier. Schott's Söhne 1995. Mainz, Germany ISBN 3-7957-5140-3.

Eine praktische Einführung zum Spielen von Jazz auf dem Klavier. Eins der ersten Kapitel enthält einen Überblick über die Akkorde, die im Jazz verwendet werden.

### *Gerou 1996*

Tom Gerou and Linda Lusk, Essential Dictionary of Music Notation. Alfred Publishing, Van Nuys CA ISBN 0-88284-768-6.

Eine ausführliche, alphabetische Liste vieler Belange des Musiksatzes und der Notation; die üblichen Fälle werden behandelt.

### *Read 1968*

Gardner Read, Music Notation: A Manual of Modern Practice. Taplinger Publishing, New York (2nd edition).

Ein Klassiker für die Musiknotation.

### *Ross 1987*

Ted Ross, Teach yourself the art of music engraving and processing. Hansen House, Miami, Florida 1987.

Dieses Buch handelt vom Musiksatz, also vom professionellen Notenstich. Hier sind Anweisungen über Stempel, die Benutzung von Stiften und nationale Konventionen versammelt. Die Kapitel zu Reproduktionstechniken und der historische Überblick sind auch interessant.

### *Schirmer 2001*

The G.Schirmer/AMP Manual of Style and Usage. G.Schirmer/AMP, NY, 2001.

Dieses Handbuch setzt den Fokus auf die Herstellung von Drucksachen für den Schirmer-Verlag. Hier werden viele Details behandelt, die sich in anderen Notationshandbüchern nicht finden. Es gibt auch einen guten Überblick, was nötig ist, um Drucke in publikationstauglicher Qualität zu produzieren.

### *Stone 1980*

Kurt Stone, Music Notation in the Twentieth Century. Norton, New York 1980.



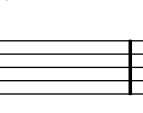
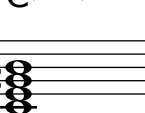
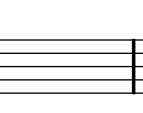
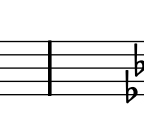



Dieses Buch enthält einen Überblick über die Notation von moderner E-Musik, beginnt aber mit einem Überblick über bereits existente Praktiken.

Das Quellenarchiv enthält eine ausführlichere Bib<sub>T</sub>E<sub>X</sub>-Bibliographie mit über 100 Titeln in 'Documentation/bibliography/"/>.

## Anhang B Notationsübersicht

### B.1 Liste der Akkordbezeichnungen

Die Tabelle zeigt die zwei üblichen Möglichkeiten, wie Akkordbezeichnungen ausgegeben werden. Es wird auch die entsprechende Note ausgegeben.

Ignatzek (default)	C	Cm	C+	C <sup>o</sup>	
Alternative	C	C <sup>b3</sup>	C <sup>#5</sup>	C <sup>b3 b5</sup>	
					
Def	C <sup>7</sup>	Cm <sup>7</sup>	C <sup>Δ</sup>	C <sup>o7</sup>	Cm <sup>Δ/b5</sup>
Alt <sub>5</sub>	C <sup>7</sup>	C <sup>7 b3</sup>	C <sup>#7</sup>	C <sup>b3 b5 b7</sup>	C <sup>b3 b5 #7</sup>
					
Def	C <sup>7/#5</sup>	Cm <sup>Δ</sup>	C <sup>Δ/#5</sup>	C <sup>ø</sup>	
Alt <sub>6</sub>	C <sup>7 #5</sup>	C <sup>b3 #7</sup>	C <sup>#5 #7</sup>	C <sup>7 b3 b5</sup>	
					
Def	C <sup>6</sup>	Cm <sup>6</sup>	C <sup>9</sup>	Cm <sup>9</sup>	
Alt <sub>4</sub>	C <sup>6</sup>	C <sup>b3 6</sup>	C <sup>9</sup>	C <sup>9 b3</sup>	
					
Def	Cm <sup>13</sup>	Cm <sup>11</sup>	Cm <sup>7/b5/9</sup>	C <sup>7/b9</sup>	
Alt <sub>8</sub>	C <sup>13 b3</sup>	C <sup>11 b3</sup>	C <sup>9 b3 b5</sup>	C <sup>7 b9</sup>	
					
Def	C <sup>7/#9</sup>	C <sup>11</sup>	C <sup>7/#11</sup>	C <sup>13</sup>	
Alt <sub>22</sub>	C <sup>7 #9</sup>	C <sup>11</sup>	C <sup>9 #11</sup>	C <sup>13</sup>	
					

Def  $C^{7/\#11/b13}$   $C^{7/\#5/\#9}$   $C^{7/\#9/\#11}$   $C^{7/b13}$   
 Alt  $C^9 \#11 \flat 13$   $C^7 \#5 \#9$   $C^7 \#9 \#11$   $C^{11 \flat 13}$   
 26

Def  $C^{7/b9/b13}$   $C^{7/\#11}$   $C^{\triangle/9}$   $C^{7/b13}$   
 Alt  $C^{11 \flat 9 \flat 13}$   $C^9 \#11$   $C^9 \#7$   $C^{11 \flat 13}$   
 30

Def  $C^{7/b9/b13}$   $C^{7/b9/13}$   $C^{\triangle/9}$   $C^{\triangle/13}$   
 Alt  $C^{11 \flat 9 \flat 13}$   $C^{13 \flat 9}$   $C^9 \#7$   $C^{13 \#7}$   
 34

Def  $C^{\triangle/\#11}$   $C^{7/b9/13}$   $C^{sus4}$   $C^{7/sus4}$   
 Alt  $C^9 \#7 \#11$   $C^{13 \flat 9}$   $C^{add4 \ 5}$   $C^{add4 \ 5 \ 7}$   
 38

Def  $C^{9/sus4}$   $C^{add9}$   $C^{m \ add11}$   
 Alt  $C^{add4 \ 5 \ 7 \ 9}$   $C^{add9}$   $C^{\flat 3 \ add11}$   
 42

## B.2 MIDI-Instrumente

Hier eine Liste von Musikinstrumentenbezeichnungen, die als Name für `midiInstrument` benutzt werden können.

acoustic grand  
 bright acoustic  
 electric grand  
 honky-tonk  
 electric piano 1  
 electric piano 2  
 harpsichord  
 clav  
 celesta  
 glockenspiel  
 music box  
 vibraphone

contrabass  
 tremolo strings  
 pizzicato strings  
 orchestral strings  
 timpani  
 string ensemble 1  
 string ensemble 2  
 synthstrings 1  
 synthstrings 2  
 choir aahs  
 voice oohs  
 synth voice

lead 7 (fifths)  
 lead 8 (bass+lead)  
 pad 1 (new age)  
 pad 2 (warm)  
 pad 3 (polysynth)  
 pad 4 (choir)  
 pad 5 (bowed)  
 pad 6 (metallic)  
 pad 7 (halo)  
 pad 8 (sweep)  
 fx 1 (rain)  
 fx 2 (soundtrack)



marimba	orchestra hit	fx 3 (crystal)
xylophone	trumpet	fx 4 (atmosphere)
tubular bells	trombone	fx 5 (brightness)
dulcimer	tuba	fx 6 (goblins)
drawbar organ	muted trumpet	fx 7 (echoes)
percussive organ	french horn	fx 8 (sci-fi)
rock organ	brass section	sitar
church organ	synthbrass 1	banjo
reed organ	synthbrass 2	shamisen
accordion	soprano sax	koto
harmonica	alto sax	kalimba
concertina	tenor sax	bagpipe
acoustic guitar (nylon)	baritone sax	fiddle
acoustic guitar (steel)	oboe	shanai
electric guitar (jazz)	english horn	tinkle bell
electric guitar (clean)	bassoon	agogo
electric guitar (muted)	clarinet	steel drums
overdriven guitar	piccolo	woodblock
distorted guitar	flute	taiko drum
guitar harmonics	recorder	melodic tom
acoustic bass	pan flute	synth drum
electric bass (finger)	blown bottle	reverse cymbal
electric bass (pick)	shakuhachi	guitar fret noise
fretless bass	whistle	breath noise
slap bass 1	ocarina	seashore
slap bass 2	lead 1 (square)	bird tweet
synth bass 1	lead 2 (sawtooth)	telephone ring
synth bass 2	lead 3 (calliope)	helicopter
violin	lead 4 (chiff)	applause
viola	lead 5 (charang)	gunshot
cello	lead 6 (voice)	

## B.3 Liste der Farben

### Normale Farben

Die Syntax zur Benutzung findet sich im Abschnitt [\[Farbige Objekte\]](#), Seite 57.

black	white	red	green
blue	cyan	magenta	yellow
grey	darkred	darkgreen	darkblue
darkcyan	darkmagenta	darkyellow	

### X-Farbbezeichnungen

X-Farbbezeichnungen haben verschiedene Varianten:

Alle Bezeichnungen, die als einziges Wort mit Großbuchstaben geschrieben werden (bspw. ‚LightSlateBlue‘), können auch von Leerzeichen getrennt geschrieben werden (also ‚light slate blue‘).

Das Wort ‚grey‘ kann in jedem Fall auch ‚gray‘ geschrieben werden (bspw. ‚DarkSlateGray‘).

Manche Bezeichnungen können auch ein numerales Suffix tragen (etwa ‚LightSalmon4‘).

### Farben ohne eine numerale Endung

snow	GhostWhite	WhiteSmoke	gainsboro	FloralWhite
------	------------	------------	-----------	-------------

OldLace	linen	AntiqueWhite	PapayaWhip	BlanchedAlmond
bisque	PeachPuff	NavajoWhite	moccasin	cornsilk
ivory	LemonChiffon	seashell	honeydew	MintCream
azure	AliceBlue	lavender	LavenderBlush	MistyRose
white	black	DarkSlateGrey	DimGrey	SlateGrey
LightSlateGrey	grey	LightGrey	MidnightBlue	navy
NavyBlue	CornflowerBlue	DarkSlateBlue	SlateBlue	MediumSlateBlue
LightSlateBlue	MediumBlue	RoyalBlue	blue	DodgerBlue
DeepSkyBlue	SkyBlue	LightSkyBlue	SteelBlue	LightSteelBlue
LightBlue	PowderBlue	PaleTurquoise	DarkTurquoise	MediumTurquoise
turquoise	cyan	LightCyan	CadetBlue	MediumAquamarine
aquamarine	DarkGreen	DarkOliveGreen	DarkSeaGreen	SeaGreen
MediumSeaGreen	LightSeaGreen	PaleGreen	SpringGreen	LawnGreen
green	chartreuse	MediumSpringGreen	GreenYellow	LimeGreen
YellowGreen	ForestGreen	OliveDrab	DarkKhaki	khaki
PaleGoldenrod	LightGoldenrodYellow	LightYellow	yellow	gold
LightGoldenrod	goldenrod	DarkGoldenrod	RosyBrown	IndianRed
SaddleBrown	sienna	peru	burlywood	beige
wheat	SandyBrown	tan	chocolate	firebrick
brown	DarkSalmon	salmon	LightSalmon	orange
DarkOrange	coral	LightCoral	tomato	OrangeRed
red	HotPink	DeepPink	pink	LightPink
PaleVioletRed	maroon	MediumVioletRed	VioletRed	magenta
violet	plum	orchid	MediumOrchid	DarkOrchid
DarkViolet	BlueViolet	purple	MediumPurple	thistle
DarkGrey	DarkBlue	DarkCyan	DarkMagenta	DarkRed
LightGreen				

## Farben mit einer numeralen Endung

Für die folgenden Bezeichnungen kann das Suffix N durch eine Zahl von 1–4 ersetzt werden.

snowN	seashellN	AntiqueWhiteN	bisqueN	PeachPuffN
NavajoWhiteN	LemonChiffonN	cornsilkN	ivoryN	honeydewN
LavenderBlushN	MistyRoseN	azureN	SlateBlueN	RoyalBlueN
blueN	DodgerBlueN	SteelBlueN	DeepSkyBlueN	SkyBlueN
LightSkyBlueN	LightSteelBlueN	LightBlueN	LightCyanN	PaleTurquoiseN
CadetBlueN	turquoiseN	cyanN	aquamarineN	DarkSeaGreenN
SeaGreenN	PaleGreenN	SpringGreenN	greenN	chartreuseN
OliveDrabN	DarkOliveGreenN	khakiN	LightGoldenrodN	LightYellowN
yellowN	goldN	goldenrodN	DarkGoldenrodN	RosyBrownN
IndianRedN	siennaN	burlywoodN	wheatN	tanN
chocolateN	firebrickN	brownN	salmonN	LightSalmonN
orangeN	DarkOrangeN	coralN	tomatoN	OrangeRedN
redN	DeepPinkN	HotPinkN	pinkN	LightPinkN
PaleVioletRedN	maroonN	VioletRedN	magentaN	orchidN
plumN	MediumOrchidN	DarkOrchidN	purpleN	MediumPurpleN
thistleN				

## Grauskala

Eine Grauskala kann mit der Bezeichnung

greyN

erstellt werden, wobei N eine Zahl von 0–100 darstellt.

## B.4 Die Feta-Schriftart

Die folgenden Symbole sind als Emmentaler-Schriftart verfügbar; auf sie kann direkt zugegriffen werden, indem man die übliche Textbeschriftung benutzt. `\musicglyph` greift direkt auf die Notationsschriftart zu (bspw. `g^{\markup { \musicglyph #"scripts.segno" }}`). Siehe auch [Abschnitt 1.8.2 \[Text formatieren\]](#), Seite 58.



space		plus	+
comma	,	hyphen	-
period	.	zero	<b>0</b>
one	<b>1</b>	two	<b>2</b>
three	<b>3</b>	four	<b>4</b>
five	<b>5</b>	six	<b>6</b>
seven	<b>7</b>	eight	<b>8</b>
nine	<b>9</b>	f	<b><i>f</i></b>
m	<b><i>m</i></b>	p	<b><i>p</i></b>
r	<b><i>r</i></b>	s	<b><i>s</i></b>
z	<b><i>z</i></b>	rests.0	<b>—</b>
rests.1	<b>—</b>	rests.0o	<b>—</b>
rests.1o	<b>—</b>	rests.M3	<b>  </b>

rests.M2	█	rests.M1	■
rests.2	↯	rests.2classical	↯
rests.3	↯	rests.4	↯
rests.5	↯	rests.6	↯
rests.7	↯	accidentals.sharp	#
accidentals.sharp .slashslash.stem	‡	accidentals.sharp .slashslashslash.stemstem	#
accidentals.sharp .slashslashslash.stem	‡	accidentals.sharp .slashslash.stemstemstem	##
accidentals.natural	♮	accidentals.flat	♭
accidentals.flat.slash	♭	accidentals.flat .slashslash	♭
accidentals .mirroredflat.flat	♭	accidentals.mirroredflat	♭
accidentals .mirroredflat.backslash	♭	accidentals.flatflat	♭♭
accidentals .flatflat.slash	♭♭	accidentals.doublsharp	×
accidentals.rightparen	)	accidentals.leftparen	(
arrowheads.open.01	>	arrowheads.open.0M1	<
arrowheads.open.11	^	arrowheads.open.1M1	∨































arrowheads.close.01	➤	arrowheads.close.0M1	➤
arrowheads.close.11	▲	arrowheads.close.1M1	▼
dots.dot	.	noteheads.uM2	⦿
noteheads.dM2	⦿	noteheads.sM1	⦿
noteheads.s0	⦿	noteheads.s1	⦿
noteheads.s2	●	noteheads.s0diamond	⦿
noteheads.s1diamond	◊	noteheads.s2diamond	◊
noteheads.s0triangle	➤	noteheads.d1triangle	➤
noteheads.ultriangle	➤	noteheads.u2triangle	➤
noteheads.d2triangle	➤	noteheads.s0slash	◊
noteheads.s1slash	◊	noteheads.s2slash	/
noteheads.s0cross	⊗	noteheads.s1cross	⊗
noteheads.s2cross	×	noteheads.s2xcircle	⊗
noteheads.s0do	△	noteheads.d1do	△
noteheads.uldo	△	noteheads.d2do	▲

noteheads.u2do	▲	noteheads.s0re	◐
noteheads.u1re	◐	noteheads.d1re	◐
noteheads.u2re	◑	noteheads.d2re	◑
noteheads.s0mi	◊	noteheads.s1mi	◊
noteheads.s2mi	◆	noteheads.u0fa	◑
noteheads.d0fa	◑	noteheads.u1fa	◑
noteheads.d1fa	◑	noteheads.u2fa	◑
noteheads.d2fa	◑	noteheads.s0la	◐
noteheads.s1la	◐	noteheads.s2la	■
noteheads.s0ti	◊	noteheads.ulti	◊
noteheads.d1ti	◊	noteheads.u2ti	◆
noteheads.d2ti	◆	scripts.ufermata	◑
scripts.dfermata	◑	scripts.ushortfermata	▲
scripts.dshortfermata	▼	scripts.ulongfermata	◐
scripts.dlongfermata	◐	scripts.uverylongfermata	◐

scripts.dverylongfermata		scripts.thumb	
scripts.sforzato		scripts.espr	
scripts.staccato		scripts.ustaccatissimo	
scripts.dstaccatissimo		scripts.tenuto	
scripts.upartato		scripts.dpartato	
scripts.umarcato		scripts.dmarcato	
scripts.open		scripts.stopped	
scripts.upbow		scripts.downbow	
scripts.reverseturn		scripts.turn	
scripts.trill		scripts.upedalheel	
scripts.dpedalheel		scripts.upedaltoe	
scripts.dpedaltoe		scripts.flageolet	
scripts.segno		scripts.coda	
scripts.varcoda		scripts.rcomma	
scripts.lcomma		scripts.rvarcomma	































<code>scripts.lvarcomma</code>		<code>scripts.arpeggio</code>	
<code>scripts.trill_element</code>		<code>scripts.arpeggio .arrow.M1</code>	
<code>scripts.arpeggio.arrow.1</code>		<code>scripts.trilelement</code>	
<code>scripts.prall</code>		<code>scripts.mordent</code>	
<code>scripts.prallprall</code>		<code>scripts.prallmordent</code>	
<code>scripts.upprall</code>		<code>scripts.upmordent</code>	
<code>scripts.pralldown</code>		<code>scripts.downprall</code>	
<code>scripts.downmordent</code>		<code>scripts.prallup</code>	
<code>scripts.lineprall</code>		<code>scripts.caesura.curved</code>	
<code>scripts.caesura.straight</code>		<code>flags.u3</code>	
<code>flags.u4</code>		<code>flags.u5</code>	
<code>flags.u6</code>		<code>flags.d3</code>	
<code>flags.ugrace</code>		<code>flags.dgrace</code>	
<code>flags.d4</code>		<code>flags.d5</code>	

































flags.d6		clefs.C	
clefs.C_change		clefs.F	
clefs.F_change		clefs.G	
clefs.G_change		clefs.percussion	
clefs.percussion_change		clefs.tab	
clefs.tab_change		timesig.C44	
timesig.C22		pedal.*	
pedal.M		pedal..	
pedal.P		pedal.d	
pedal.e		pedal.Ped	
brackettips.up		brackettips.down	
accordion.accDiscant		accordion.accDot	
accordion.accFreebase		accordion.accStdbase	
accordion.accBayanbase		accordion.accOldEE	
rests.M3neomensural		rests.M2neomensural	































rests.M1neomensural	⏏	rests.0neomensural	⏏
rests.1neomensural	⏏	rests.2neomensural	⏏
rests.3neomensural	⏏	rests.4neomensural	⏏
rests.M3mensural	⏏	rests.M2mensural	⏏
rests.M1mensural	⏏	rests.0mensural	⏏
rests.1mensural	⏏	rests.2mensural	⏏
rests.3mensural	⏏	rests.4mensural	⏏
noteheads.s1neomensural	⏏	noteheads.sM3neomensural	⏏
noteheads.sM2neomensural	⏏	noteheads.sM1neomensural	⏏
noteheads.s0harmonic	⏏	noteheads.s2harmonic	⏏
noteheads.s0neomensural	⏏	noteheads.s1neomensural	⏏
noteheads.s2neomensural	⏏	noteheads.s1mensural	⏏
noteheads.sM3mensural	⏏	noteheads.sM2mensural	⏏
noteheads.sM1mensural	⏏	noteheads.s0mensural	⏏
noteheads.s1mensural	⏏	noteheads.s2mensural	⏏

noteheads.s0petrucci	◊	noteheads.s1petrucci	◊
noteheads.s2petrucci	◆	noteheads .svaticana.punctum	■
noteheads.svaticana .punctum.cavum	◻	noteheads.svaticana .linea.punctum	▣
noteheads.svaticana .linea.punctum.cavum	◻	noteheads.svaticana .inclinatum	◆
noteheads.svaticana.lpes	■	noteheads .svaticana.vlpes	■
noteheads.svaticana.upes	■	noteheads .svaticana.vupes	■
noteheads .svaticana.plica	·	noteheads .svaticana.vplica	·
noteheads .svaticana.epiphonus	┐	noteheads.svaticana .vepiphonus	┐
noteheads.svaticana .reverse.plica	·	noteheads.svaticana .reverse.vplica	·
noteheads.svaticana .inner.cephalicus	┐	noteheads.svaticana .cephalicus	┐
noteheads .svaticana.quilisma	┐	noteheads.ssolesmes .incl.parvum	·
noteheads .ssolesmes.auct.asc	┐	noteheads .ssolesmes.auct.desc	┐
noteheads.ssolesmes .incl.auctum	┐	noteheads .ssolesmes.stropha	┐
noteheads.ssolesmes .stropha.aucta	┐	noteheads .ssolesmes.oriscus	┐
noteheads.smedicaea .inclinatum	◆	noteheads .smedicaea.punctum	■

noteheads .smedicaea.rvirga		noteheads .smedicaea.virga	
noteheads .shufnagel.punctum		noteheads .shufnagel.virga	
noteheads.shufnagel.lpes		clefs.vaticana.do	
clefs.vaticana.do_change		clefs.vaticana.fa	
clefs.vaticana.fa_change		clefs.medicaea.do	
clefs.medicaea.do_change		clefs.medicaea.fa	
clefs.medicaea.fa_change		clefs.neomensural.c	
clefs.neomensural .c_change		clefs.petrucchi.c1	
clefs.petrucchi.c1_change		clefs.petrucchi.c2	
clefs.petrucchi.c2_change		clefs.petrucchi.c3	
clefs.petrucchi.c3_change		clefs.petrucchi.c4	
clefs.petrucchi.c4_change		clefs.petrucchi.c5	
clefs.petrucchi.c5_change		clefs.mensural.c	
clefs.mensural.c_change		clefs.petrucchi.f	
clefs.petrucchi.f_change		clefs.mensural.f	

clefs.mensural.f_change		clefs.petrucchi.g	
clefs.petrucchi.g_change		clefs.mensural.g	
clefs.mensural.g_change		clefs.hufnagel.do	
clefs.hufnagel.do_change		clefs.hufnagel.fa	
clefs.hufnagel.fa_change		clefs.hufnagel.do.fa	
clefs.hufnagel .do.fa_change		custodes.hufnagel.u0	
custodes.hufnagel.u1		custodes.hufnagel.u2	
custodes.hufnagel.d0		custodes.hufnagel.d1	
custodes.hufnagel.d2		custodes.medicaea.u0	
custodes.medicaea.u1		custodes.medicaea.u2	
custodes.medicaea.d0		custodes.medicaea.d1	
custodes.medicaea.d2		custodes.vaticana.u0	
custodes.vaticana.u1		custodes.vaticana.u2	
custodes.vaticana.d0		custodes.vaticana.d1	
custodes.vaticana.d2		custodes.mensural.u0	

custodes.mensural.u1	↗	custodes.mensural.u2	↗
custodes.mensural.d0	↘	custodes.mensural.d1	↘
custodes.mensural.d2	↘	accidentals.medicaeaM1	♭
accidentals.vaticanaM1	♭	accidentals.vaticana0	♯
accidentals.mensural1	✕	accidentals.mensuralM1	♭
accidentals.hufnagelM1	♭	flags.mensuralu03	}
flags.mensuralu13	)	flags.mensuralu23	)
flags.mensurald03	(	flags.mensurald13	(
flags.mensurald23	(	flags.mensuralu04	}
flags.mensuralu14	}	flags.mensuralu24	}
flags.mensurald04	{	flags.mensurald14	{
flags.mensurald24	{	flags.mensuralu05	}
flags.mensuralu15	}	flags.mensuralu25	}
flags.mensurald05	{	flags.mensurald15	{
flags.mensurald25	{	flags.mensuralu06	}

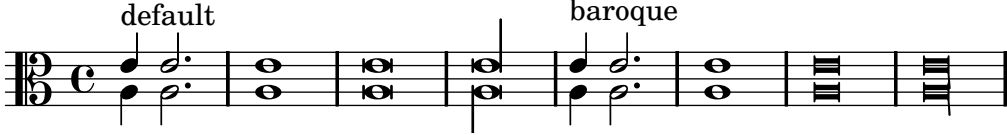
flags.mensuralu16		flags.mensuralu26	
flags.mensurald06		flags.mensurald16	
flags.mensurald26		timesig.mensural44	
timesig.mensural22		timesig.mensural32	
timesig.mensural64		timesig.mensural94	
timesig.mensural34		timesig.mensural68	
timesig.mensural98		timesig.mensural48	
timesig.mensural68alt		timesig.mensural24	
timesig.neomensural44		timesig.neomensural22	
timesig.neomensural32		timesig.neomensural64	
timesig.neomensural94		timesig.neomensural34	
timesig.neomensural68		timesig.neomensural98	
timesig.neomensural48		timesig.neomensural68alt	
timesig.neomensural24		scripts.ictus	
scripts.uaccentus		scripts.daccentus	

<code>scripts.usemicirculus</code>	.	<code>scripts.dsemicirculus</code>	.
<code>scripts.circulus</code>	◦	<code>scripts.augmentum</code>	.
<code>scripts</code> <code>.usignumcongruentiae</code>	§	<code>scripts</code> <code>.dsignumcongruentiae</code>	§
<code>dots.dotvaticana</code>	.		


## B.5 Notenkopfstile

Folgende Stile können zur Darstellung der Notenköpfe verwendet werden:


default                      baroque




9                      neomensural                      mensural



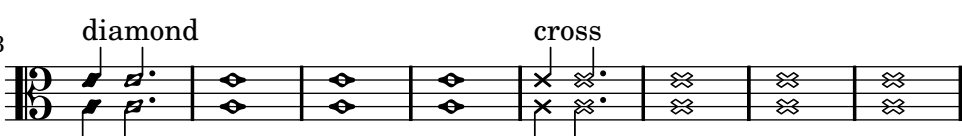
17                      petrucci                      harmonic



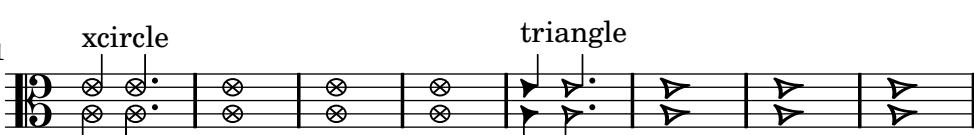
25                      harmonic-black                      harmonic-mixed




33                      diamond                      cross



41                      xcircle                      triangle



49                      slash





## B.6 Text markup commands

The following commands can all be used inside `\markup { }`.

### B.6.1 Font

`\abs-fontsize size (number) arg (markup)`

Use *size* as the absolute font size to display *arg*. Adjust baseline skip and word space accordingly.

```
\markup {
  default text font size
  \hspace #2
  \abs-fontsize #16 { text font size 16 }
  \hspace #2
  \abs-fontsize #12 { text font size 12 }
}
```

**default text font size    text font size 16    text font size 12**

`\bigger arg (markup)`

Increase the font size relative to current setting.

```
\markup {
  \huge {
    huge
    \hspace #2
    \bigger {
      bigger
    }
    \hspace #2
    huge
  }
}
```

**huge    bigger    huge**

`\bold arg (markup)`

Switch to bold font-series.

```
\markup {
  default
  \hspace #2
  \bold
  bold
}
```

**default    bold**

`\box arg (markup)`

Draw a box round *arg*. Looks at **thickness**, **box-padding** and **font-size** properties to determine line thickness and padding around the markup.

```
\markup {
  \override #'(box-padding . 0.5)
  \box
```

```
\line { V. S. }
}
```

**V. S.**

Used properties:

- `box-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\caps arg` (markup)

Copy of the `\smallCaps` command.

```
\markup {
  default
  \hspace #2
  \caps {
    Text in small caps
  }
}
```

**default    TEXT IN SMALL CAPS**

`\dynamic arg` (markup)

Use the dynamic font. This font only contains **s**, **f**, **m**, **z**, **p**, and **r**. When producing phrases, like ,più **f**, the normal words (like ,più‘) should be done in a different font. The recommended font for this is bold and italic.

```
\markup {
  \dynamic {
    sfzp
  }
}
```

***sfzp***

`\finger arg` (markup)

Set the argument as small numbers.

```
\markup {
  \finger {
    1 2 3 4 5
  }
}
```

**1 2 3 4 5**

`\fontCaps arg` (markup)

Set `font-shape` to `caps`

Note: `\fontCaps` requires the installation and selection of fonts which support the `caps` font shape.

`\fontsize increment` (number) `arg` (markup)

Add *increment* to the font-size. Adjust baseline skip accordingly.

```
\markup {
  default
  \hspace #2
  \fontsize #-1.5
  smaller
}
```

**default**    **smaller**

Used properties:

- `baseline-skip` (2)
- `word-space` (1)
- `font-size` (0)

`\huge arg` (markup)  
Set font size to +2.

```
\markup {
  default
  \hspace #2
  \huge
  huge
}
```

**default**    **huge**

`\italic arg` (markup)  
Use italic `font-shape` for `arg`.

```
\markup {
  default
  \hspace #2
  \italic
  italic
}
```

**default**    ***italic***

`\large arg` (markup)  
Set font size to +1.

```
\markup {
  default
  \hspace #2
  \large
  large
}
```

**default**    **large**

`\larger arg` (markup)  
Copy of the `\bigger` command.

```
\markup {
  default
  \hspace #2
  \larger
  larger
}
```

**default    larger**

`\magnify sz (number) arg (markup)`

Set the font magnification for its argument. In the following example, the middle A is 10% larger:

```
A \magnify #1.1 { A } A
```

Note: Magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.

```
\markup {
  default
  \hspace #2
  \magnify #1.5 {
    50% larger
  }
}
```

**default    50% larger**

`\medium arg (markup)`

Switch to medium font series (in contrast to bold).

```
\markup {
  \bold {
    some bold text
    \hspace #2
    \medium {
      medium font series
    }
    \hspace #2
    bold again
  }
}
```

**some bold text    medium font series    bold again**

`\normal-size-sub arg (markup)`

Set *arg* in subscript, in a normal font size.

```
\markup {
  default
  \normal-size-sub {
    subscript in standard size
  }
}
```

**default**  
subscript in standard size

Used properties:

- `baseline-skip`

`\normal-size-super arg` (markup)

Set *arg* in superscript with a normal font size.

```
\markup {
  default
  \normal-size-super {
    superscript in standard size
  }
}
```

**default** **superscript in standard size**

Used properties:

- `baseline-skip`

`\normal-text arg` (markup)

Set all font related properties (except the size) to get the default normal text font, no matter what font was used earlier.

```
\markup {
  \huge \bold \sans \caps {
    Some text with font overrides
    \hspace #2
    \normal-text {
      Default text, same font-size
    }
    \hspace #2
    More text as before
  }
}
```

**SOME TEXT WITH FONT OVERRIDES**    Default text, same font-size    **MORE**

`\normalsize arg` (markup)

Set font size to default.

```
\markup {
  \teeny {
    this is very small
    \hspace #2
    \normalsize {
      normal size
    }
    \hspace #2
    teeny again
  }
}
```

this is very small    **normal size**    teeny again

`\number arg` (markup)

Set font family to `number`, which yields the font used for time signatures and fingerings. This font only contains numbers and some punctuation. It doesn't have any letters.

```
\markup {
  \number {
    0 1 2 3 4 5 6 7 8 9 . ,
  }
}
```

**0123456789.,**

`\roman arg` (markup)

Set font family to `roman`.

```
\markup {
  \sans \bold {
    sans serif, bold
    \hspace #2
    \roman {
      text in roman font family
    }
    \hspace #2
    return to sans
  }
}
```

**sans serif, bold    text in roman font family    return to sans**

`\sans arg` (markup)

Switch to the sans serif family.

```
\markup {
  default
  \hspace #2
  \sans {
    sans serif
  }
}
```

**default    sans serif**

`\simple str` (string)

A simple text string; `\markup { foo }` is equivalent with `\markup { \simple #"foo" }`.

Note: for creating standard text markup or defining new markup commands, the use of `\simple` is unnecessary.

```
\markup {
  \simple #"simple"
  \simple #"text"
  \simple #"strings"
}
```

**simple text strings**`\small arg (markup)`

Set font size to -1.

```

\markup {
  default
  \hspace #2
  \small
  small
}

```

**default    small**`\smallCaps text (markup)`Emit *arg* as small caps.Note: `\smallCaps` does not support accented characters.

```

\markup {
  default
  \hspace #2
  \smallCaps {
    Text in small caps
  }
}

```

**default    TEXT IN SMALL CAPS**`\smaller arg (markup)`

Decrease the font size relative to current setting.

```

\markup {
  \fontsize #3.5 {
    some large text
    \hspace #2
    \smaller {
      a bit smaller
    }
    \hspace #2
    more large text
  }
}

```

**some large text    a bit smaller    more large text**`\sub arg (markup)`Set *arg* in subscript.

```

\markup {
  \concat {
    H
    \sub {
      2
    }
  }
  0
}

```

```

    }
  }

```

$$\text{H}_2\text{O}$$

Used properties:

- `baseline-skip`
- `font-size (0)`

`\super arg` (markup)

Raising and lowering texts can be done with `\super` and `\sub`:

```

\markup {
  E =
  \concat {
    mc
    \super
    2
  }
}

```

$$E = mc^2$$

Used properties:

- `baseline-skip`
- `font-size (0)`

`\teeny arg` (markup)

Set font size to -3.

```

\markup {
  default
  \hspace #2
  \teeny
  teeny
}

```

**default**      *teeny*

`\text arg` (markup)

Use a text font instead of music symbol or music alphabet font.

```

\markup {
  \number {
    1, 2,
    \text {
      three, four,
    }
    5
  }
}

```

**1, 2**, three, four, **5**

`\tiny arg` (markup)

Set font size to -2.



```
\markup {
  default
  \hspace #2
  \tiny
  tiny
}
```

**default**    *tiny*

`\typewriter` *arg* (markup)  
Use font-family typewriter for *arg*.

```
\markup {
  default
  \hspace #2
  \typewriter
  typewriter
}
```

**default**    **typewriter**

`\underline` *arg* (markup)  
Underline *arg*. Looks at **thickness** to determine line thickness and y offset.

```
\markup {
  default
  \hspace #2
  \override #'(thickness . 2)
  \underline {
    underline
  }
}
```

**default**    **underline**

Used properties:

- **thickness** (1)

`\upright` *arg* (markup)  
Set font shape to **upright**. This is the opposite of *italic*.

```
\markup {
  \italic {
    italic text
    \hspace #2
    \upright {
      upright text
    }
  }
  \hspace #2
  italic again
}
```

*italic text*    **upright text**    *italic again*

## B.6.2 Align

`\center-align` *arg* (markup)

Align *arg* to its X center.

```
\markup {
  \column {
    one
    \center-align
    two
    three
  }
}
```

one  
two  
three

`\center-column` *args* (list of markups)

Put *args* in a centered column.

```
\markup {
  \center-column {
    one
    two
    three
  }
}
```

one  
two  
three

Used properties:

- `baseline-skip`

`\column` *args* (list of markups)

Stack the markups in *args* vertically. The property `baseline-skip` determines the space between each markup in *args*.

```
\markup {
  \column {
    one
    two
    three
  }
}
```

one  
two  
three

Used properties:

- `baseline-skip`

`\combine` *m1* (markup) *m2* (markup)

Print two markups on top of each other.

Note: `\combine` cannot take a list of markups enclosed in curly braces as an argument; the follow example will not compile:

```
\combine { a list }
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \combine
    \draw-line #'(0 . 4)
    \arrow-head #Y #DOWN ##f
}
```



`\concat` *args* (list of markups)

Concatenate *args* in a horizontal line, without spaces inbetween. Strings and simple markups are concatenated on the input level, allowing ligatures. For example, `\concat { "f" \simple #"i" }` is equivalent to `"fi"`.

```
\markup {
  \concat {
    one
    two
    three
  }
}
```

**onetwothree**

`\dir-column` *args* (list of markups)

Make a column of *args*, going up or down, depending on the setting of the  `#'direction` layout property.

```
\markup {
  \override #'(direction . 1) {
    \dir-column {
      going up
    }
  }
  \dir-column {
    going down
  }
}
```

**up**  
going going  
down

Used properties:

- `baseline-skip`
- `direction`

`\fill-line` *markups* (list of markups)

Put *markups* in a horizontal line of width *line-width*. The markups are spaced or flushed to fill the entire line. If there are no arguments, return an empty stencil.

```
\markup {
  \column {
    \fill-line {
      Words evenly spaced across the page
    }
    \null
    \fill-line {
      \line { Text markups }
      \line {
        \italic { evenly spaced }
      }
      \line { across the page }
    }
  }
}
```

Words evenly spaced across the page

Text markups *evenly spaced* across the page

Used properties:

- `line-width` (#f)
- `word-space` (1)
- `text-direction` (1)

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)

Align *arg* in *axis* direction to the *dir* side.

```
\markup {
  \column {
    one
    \general-align #X #LEFT
    two
    three
  }
  \null
  one
  \general-align #X #CENTER
  two
  three
  \null
  \line {
    one
    \general-align #Y #UP
    two
    three
  }
  \null
  \line {
    one
```

```

        \general-align #Y #3.2
        two
        three
    }
}

```

```

one
two
three

```

```

one
two
three

```

```

one two three

```

```

one three
two

```

`\halign` *dir* (number) *arg* (markup)

Set horizontal alignment. If *dir* is -1, then it is left-aligned, while +1 is right. Values in between interpolate alignment accordingly.

```

\markup {
  \column {
    one
    \halign #LEFT
    two
    three
    \null
    one
    \halign #CENTER
    two
    three
    \null
    one
    \halign #RIGHT
    two
    three
    \null
    one
    \halign #-5
    two
    three
  }
}

```

one  
two  
three

one  
two  
three

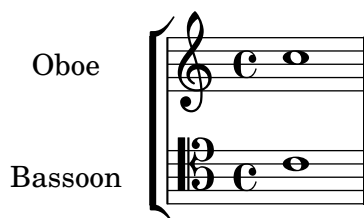
one  
two  
three

one  
two  
three

`\hcenter-in` *length* (number) *arg* (markup)

Center *arg* horizontally within a box of extending *length*/2 to the left and right.

```
\new StaffGroup <<
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Oboe
    }
    c'1
  }
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Bassoon
    }
    \clef tenor
    c'1
  }
>>
```



`\hspace` *amount* (number)

This produces an invisible object taking horizontal space. For example,

```
\markup { A \hspace #2.0 B }
```

puts extra space between A and B, on top of the space that is normally inserted before elements on a line.

```
\markup {
  one
  \hspace #2
  two
}
```

```

\hspace #8
three
}

one    two    three

```

`\justify-field symbol (symbol)`

Justify the data which has been assigned to *symbol*.

```

\header {
  title = "My title"
  descr = "Lorem ipsum dolor sit amet, consectetur adipisicing elit,
  sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
  Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
  nisi ut aliquip ex ea commodo consequat."
}

```

```

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \justify-field #'header:descr
    }
  }
}

```

```

\markup {
  \null
}

```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\justify args (list of markups)`

Like wordwrap, but with lines stretched to justify the margins. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```

\markup {
  \justify {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  }
}

```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\justify-string` *arg* (string)

Justify a string. Paragraphs may be separated with double newlines

```
\markup {
  \override #'(line-width . 40)
  \justify-string #"Lorem ipsum dolor sit amet, consectetur
    adipisicing elit, sed do eiusmod tempor incididunt ut labore
    et dolore magna aliqua.
```

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum"

```
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

`\left-align` *arg* (markup)

Align *arg* on its left edge.

```
\markup {
  \column {
    one
    \left-align
    two
```



```

        three
    }
}

```

```

    one
    two
    three

```

`\left-column` *args* (list of markups)  
Put *args* in a left-aligned column.

```

\markup {
  \left-column {
    one
    two
    three
  }
}

```

```

    one
    two
    three

```

Used properties:

- `baseline-skip`

`\line` *args* (list of markups)  
Put *args* in a horizontal line. The property `word-space` determines the space between each markup in *args*.

```

\markup {
  \line {
    one two three
  }
}

```

```

    one two three

```

Used properties:

- `text-direction` (1)
- `word-space`

`\lower` *amount* (number) *arg* (markup)  
Lower *arg* by the distance *amount*. A negative *amount* indicates raising; see also `\raise`.

```

\markup {
  one
  \lower #3
  two
  three
}

```

```

    one      three
      two

```

`\pad-around` *amount* (number) *arg* (markup)  
Add padding *amount* all around *arg*.

```

\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-around #0.5 {
      padded
    }
  }
}

```

default padded

`\pad-markup padding (number) arg (markup)`  
 Add space around a markup object.

```

\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-around #1 {
      padded
    }
  }
}

```

default padded

`\pad-to-box x-ext (pair of numbers) y-ext (pair of numbers) arg (markup)`  
 Make *arg* take at least *x-ext*, *y-ext* space.

```

\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-to-box #'(0 . 10) #'(0 . 3) {
      padded
    }
  }
}

```

default padded

`\pad-x amount (number) arg (markup)`  
 Add padding *amount* around *arg* in the X direction.

```

\markup {

```

```

\box {
  default
}
\hspace #4
\box {
  \pad-x #2 {
    padded
  }
}
}

```

default	padded
---------	--------

`\put-adjacent` *axis* (integer) *dir* (direction) *arg1* (markup) *arg2* (markup)  
 Put *arg2* next to *arg1*, without moving *arg1*.

`\raise` *amount* (number) *arg* (markup)

Raise *arg* by the distance *amount*. A negative *amount* indicates lowering, see also `\lower`.

The argument to `\raise` is the vertical displacement amount, measured in (global) staff spaces. `\raise` and `\super` raise objects in relation to their surrounding markups.

If the text object itself is positioned above or below the staff, then `\raise` cannot be used to move it, since the mechanism that positions it next to the staff cancels any shift made with `\raise`. For vertical positioning, use the `padding` and/or `extra-offset` properties.

```

\markup {
  C
  \small
  \bold
  \raise #1.0
  9/7+
}

```

**C 9/7+**

`\right-align` *arg* (markup)

Align *arg* on its right edge.

```

\markup {
  \column {
    one
    \right-align
    two
    three
  }
}

```

one  
 two  
 three

`\right-column` *args* (list of markups)

Put *args* in a right-aligned column.

```
\markup {
  \right-column {
    one
    two
    three
  }
}
```

one  
two  
three

Used properties:

- `baseline-skip`

`\rotate` *ang* (number) *arg* (markup)

Rotate object with *ang* degrees around its center.

```
\markup {
  default
  \hspace #2
  \rotate #45
  \line {
    rotated 45°
  }
}
```

default

rotated 45°

`\translate` *offset* (pair of numbers) *arg* (markup)

This translates an object. Its first argument is a cons of numbers.

```
A \translate #(cons 2 -3) { B C } D
```

This moves ,B C' 2 spaces to the right, and 3 down, relative to its surroundings.

This command cannot be used to move isolated scripts vertically, for the same reason that `\raise` cannot be used for that.

```
\markup {
  *
  \translate #'(2 . 3)
  \line { translated two spaces right, three up }
}
```

translated two spaces right, three up

\*

`\translate-scaled` *offset* (pair of numbers) *arg* (markup)

Translate *arg* by *offset*, scaling the offset by the `font-size`.

```
\markup {
  \fontsize #5 {
    * \translate #'(2 . 3) translate
    \hspace #2
  }
}
```

```

    * \translate-scaled #'(2 . 3) translate-scaled
  }
}

```

$\ast$  **translate**  $\ast$  **translate-scaled**

Used properties:

- font-size (0)

`\vcenter` *arg* (markup)

Align *arg* to its Y center.

```

\markup {
  one
  \vcenter
  two
  three
}

```

one two three

`\wordwrap-field` *symbol* (symbol)

Wordwrap the data which has been assigned to *symbol*.

```

\header {
  title = "My title"
  descr = "Lorem ipsum dolor sit amet, consectetur adipisicing elit,
  sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
  Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
  nisi ut aliquip ex ea commodo consequat."
}

```

```

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \wordwrap-field #'header:descr
    }
  }
}

```

```

\markup {
  \null
}

```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod  
 tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim  
 veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea  
 commodo consequat.

`\wordwrap` *args* (list of markups)

Simple wordwrap. Use `\override #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

```
\markup {
  \wordwrap {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\wordwrap-string` *arg* (string)

Wordwrap a string. Paragraphs may be separated with double newlines.

```
\markup {
  \override #'(line-width . 40)
  \wordwrap-string #"Lorem ipsum dolor sit amet, consectetur
  adipisicing elit, sed do eiusmod tempor incididunt ut labore
  et dolore magna aliqua.

  Ut enim ad minim veniam, quis nostrud exercitation ullamco
  laboris nisi ut aliquip ex ea commodo consequat.

  Excepteur sint occaecat cupidatat non proident, sunt in culpa
  qui officia deserunt mollit anim id est laborum"
}
```

Lorem ipsum dolor sit amet,  
 consectetur adipisicing elit, sed do  
 eiusmod tempor incididunt ut labore  
 et dolore magna aliqua.  
 Ut enim ad minim veniam, quis  
 nostrud exercitation ullamco laboris  
 nisi ut aliquip ex ea commodo  
 consequat.  
 Excepteur sint occaecat cupidatat non  
 proident, sunt in culpa qui officia  
 deserunt mollit anim id est laborum

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

### B.6.3 Graphic

`\arrow-head` *axis* (integer) *direction* (direction) *filled* (boolean)

Produce an arrow head in specified direction and axis. Use the filled head if *filled* is specified.

```

\markup {
  \fontsize #5 {
    \general-align #Y #DOWN {
      \arrow-head #Y #UP ##t
      \arrow-head #Y #DOWN ##f
      \hspace #2
      \arrow-head #X #RIGHT ##f
      \arrow-head #X #LEFT ##f
    }
  }
}

```

▲Y ><

`\beam` *width* (number) *slope* (number) *thickness* (number)

Create a beam with the specified parameters.

```

\markup {
  \beam #5 #1 #2
}

```



`\bracket` *arg* (markup)

Draw vertical brackets around *arg*.

```

\markup {
  \bracket {
    \note #"2." #UP
  }
}

```

}

[J.]

`\circle arg (markup)`

Draw a circle around *arg*. Use `thickness`, `circle-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \circle {
    Hi
  }
}
```

Ⓜ

Used properties:

- `circle-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\draw-circle radius (number) thickness (number) fill (boolean)`

A circle of radius *radius*, thickness *thickness* and optionally filled.

```
\markup {
  \draw-circle #2 #0.5 ##f
  \hspace #2
  \draw-circle #2 #0 ##t
}
```



`\draw-line dest (pair of numbers)`

A simple line.

```
\markup {
  \draw-line #'(4 . 4)
  \override #'(thickness . 5)
  \draw-line #'(-3 . 0)
}
```



Used properties:

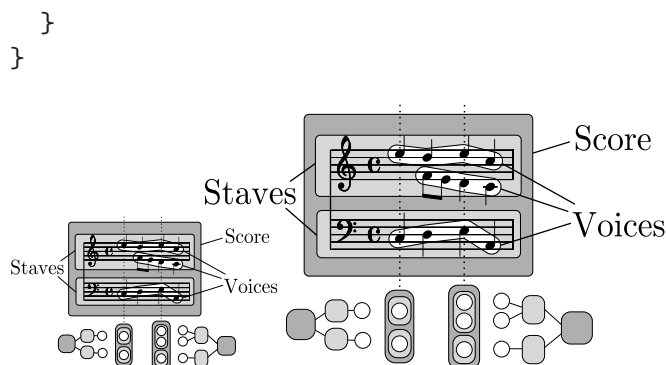
- `thickness` (1)

`\epsfile axis (number) size (number) file-name (string)`

Inline an EPS image. The image is scaled along *axis* to *size*.

```
\markup {
  \general-align #Y #DOWN {
    \epsfile #X #20 #"context-example.eps"
    \epsfile #Y #20 #"context-example.eps"
  }
}
```





`\filled-box` *xext* (pair of numbers) *yext* (pair of numbers) *blot* (number)

Draw a box with rounded corners of dimensions *xext* and *yext*. For example,

```
\filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0
```

creates a box extending horizontally from -0.3 to 1.8 and vertically from -0.3 up to 1.8, with corners formed from a circle of diameter 0 (i.e. sharp corners).

```
\markup {
  \filled-box #'(0 . 4) #'(0 . 4) #0
  \filled-box #'(0 . 2) #'(-4 . 2) #0.4
  \filled-box #'(1 . 8) #'(0 . 7) #0.2
  \with-color #white
  \filled-box #'(-4.5 . -2.5) #'(3.5 . 5.5) #0.7
}
```



`\hbracket` *arg* (markup)

Draw horizontal brackets around *arg*.

```
\markup {
  \hbracket {
    \line {
      one two three
    }
  }
}
```

one two three

`\postscript` *str* (string)

This inserts *str* directly into the output as a PostScript command string. Due to technicalities of the output backends, different scales should be used for the T<sub>E</sub>X and PostScript backend, selected with `-f`.

For the T<sub>E</sub>X backend, the following string prints a rotated text

```
0 0 moveto /ecrm10 findfont
1.75 scalefont setfont 90 rotate (hello) show
```

The magical constant 1.75 scales from LilyPond units (staff spaces) to T<sub>E</sub>X dimensions.

For the postscript backend, use the following

```

gsave /ecrm10 findfont
  10.0 output-scale div
  scalefont setfont 90 rotate (hello) show grestore

eyeglassesps = #"
  0.15 setlinewidth
  -0.9 0 translate
  1.1 1.1 scale
  1.2 0.7 moveto
  0.7 0.7 0.5 0 361 arc
  stroke
  2.20 0.70 0.50 0 361 arc
  stroke
  1.45 0.85 0.30 0 180 arc
  stroke
  0.20 0.70 moveto
  0.80 2.00 lineto
  0.92 2.26 1.30 2.40 1.15 1.70 curveto
  stroke
  2.70 0.70 moveto
  3.30 2.00 lineto
  3.42 2.26 3.80 2.40 3.65 1.70 curveto
  stroke"

eyeglasses = \markup {
  \with-dimensions #'(0 . 4.4) #'(0 . 2.5)
  \postscript #eyeglassesps
}

\relative c'' {
  c2^\eyeglasses
  a2_\eyeglasses
}

```



`\rounded-box` *arg* (markup)

Draw a box with rounded corners around *arg*. Looks at **thickness**, **box-padding** and **font-size** properties to determine line thickness and padding around the markup; the **corner-radius** property makes it possible to define another shape for the corners (default is 1).

```

c4^\markup {
  \rounded-box {
    Overtura
  }
}
c,8. c16 c4 r

```



Used properties:

- `box-padding` (0.5)
- `font-size` (0)
- `corner-radius` (1)
- `thickness` (1)

`\triangle` *filled* (boolean)

A triangle, either filled or empty.

```
\markup {
  \triangle ##t
  \hspace #2
  \triangle ##f
}
```



Used properties:

- `baseline-skip` (2)
- `font-size` (0)
- `thickness` (0.1)

`\with-url` *url* (string) *arg* (markup)

Add a link to URL *url* around *arg*. This only works in the PDF backend.

```
\markup {
  \with-url #"http://lilypond.org/web/" {
    LilyPond ... \italic {
      music notation for everyone
    }
  }
}
```

LilyPond ... *music notation for everyone*

## B.6.4 Music

`\doubleflat`

Draw a double flat symbol.

```
\markup {
  \doubleflat
}
```



`\doublesharp`

Draw a double sharp symbol.

```
\markup {
  \doublesharp
}
```

✕

`\flat`

Draw a flat symbol.

```
\markup {
  \flat
}
```

b

`\musicglyph` *glyph-name* (string)

*glyph-name* is converted to a musical symbol; for example, `\musicglyph #\"accidentals.natural\"` selects the natural sign from the music font. See [Abschnitt “Die Feta-Schriftart” in \*Benutzerhandbuch\*](#) for a complete listing of the possible glyphs.

```
\markup {
  \musicglyph #\"f\"
  \musicglyph #\"rests.2\"
  \musicglyph #\"clefs.G_change\"
}
```

*f* 

`\natural`

Draw a natural symbol.

```
\markup {
  \natural
}
```

q

`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)

Construct a note symbol, with stem. By using fractional values for *dir*, you can obtain longer or shorter stems.

```
\markup {
  \note-by-number #3 #0 #DOWN
  \hspace #2
  \note-by-number #1 #2 #0.8
}
```

 *d..*

Used properties:

- `style` ('())
- `font-size` (0)

`\note` *duration* (string) *dir* (number)

This produces a note with a stem pointing in *dir* direction, with the *duration* for the note head type and augmentation dots. For example, `\note #\"4.\" #-0.75` creates a dotted quarter note, with a shortened down stem.

```
\markup {
  \override #'(style . cross) {
    \note #"4.." #UP
  }
  \hspace #2
  \note #"breve" #0
}
```



Used properties:

- `style ('())`
- `font-size (0)`

`\score score` (unknown)

Inline an image of music.

```
\markup {
  \score {
    \new PianoStaff <<
      \new Staff \relative c' {
        \key f \major
        \time 3/4
        \mark \markup { Allegro }
        f2\p( a4)
        c2( a4)
        bes2( g'4)
        f8( e) e4 r
      }
      \new Staff \relative c {
        \clef bass
        \key f \major
        \time 3/4
        f8( a c a c a
        f c' es c es c)
        f,( bes d bes d bes)
        f( g bes g bes g)
      }
    >>
    \layout {
      indent = 0.0\cm
      \context {
        \Score
        \override RehearsalMark #'break-align-symbols =
          #'(time-signature key-signature)
        \override RehearsalMark #'self-alignment-X = #LEFT
      }
      \context {
        \Staff
        \override TimeSignature #'break-align-anchor-alignment = #LEFT
      }
    }
  }
}
```



`\semiflat`

Draw a semiflat symbol.

```
\markup {
  \semiflat
}
```



`\semisharp`

Draw a semi sharp symbol.

```
\markup {
  \semisharp
}
```



`\sesquiflat`

Draw a 3/2 flat symbol.

```
\markup {
  \sesquiflat
}
```



`\sesquisharp`

Draw a 3/2 sharp symbol.

```
\markup {
  \sesquisharp
}
```



`\sharp`

Draw a sharp symbol.

```
\markup {
  \sharp
}
```



#

`\tied-lyric str (string)`Like `simple-markup`, but use tie characters for ,~‘ tilde symbols.

```
\markup {
  \tied-lyric #"Lasciate~i monti"
}
```

Lasciate i monti

### B.6.5 Instrument Specific Markup

`\fret-diagram definition-string (string)`

Make a (guitar) fret diagram. For example, say

```
\markup \fret-diagram #"s:0.75;6-x;5-x;4-o;3-2;2-3;1-2;"
```

for fret spacing 3/4 of staff space, D chord diagram

Syntax rules for *definition-string*:

- Diagram items are separated by semicolons.
- Possible items:
  - `s: number` – Set the fret spacing of the diagram (in staff spaces). Default: 1.
  - `t: number` – Set the line thickness (in staff spaces). Default: 0.05.
  - `h: number` – Set the height of the diagram in frets. Default: 4.
  - `w: number` – Set the width of the diagram in strings. Default: 6.
  - `f: number` – Set fingering label type (0 = none, 1 = in circle on string, 2 = below string). Default: 0.
  - `d: number` – Set radius of dot, in terms of fret spacing. Default: 0.25.
  - `p: number` – Set the position of the dot in the fret space. 0.5 is centered; 1 is on lower fret bar, 0 is on upper fret bar. Default: 0.6.
  - `c: string1-string2-fret` – Include a barre mark from *string1* to *string2* on *fret*.
  - `string-fret` – Place a dot on *string* at *fret*. If *fret* is ‘o’, *string* is identified as open. If *fret* is ‘x’, *string* is identified as muted.
  - `string-fret-fingering` – Place a dot on *string* at *fret*, and label with *fingering* as defined by the `f:` code.
- Note: There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-terse definition-string (string)`

Make a fret diagram markup using terse string-based syntax.

Here is an example

```
\markup \fret-diagram-terse #"x;x;o;2;3;2;"
```

for a D chord diagram.

Syntax rules for *definition-string*:

- Strings are terminated by semicolons; the number of semicolons is the number of strings in the diagram.
- Mute strings are indicated by ‘x’.
- Open strings are indicated by ‘o’.
- A number indicates a fret indication at that fret.
- If there are multiple fret indicators desired on a string, they should be separated by spaces.
- Fingerings are given by following the fret number with a -, followed by the finger indicator, e.g. ‘3-2’ for playing the third fret with the second finger.
- Where a barre indicator is desired, follow the fret (or fingering) symbol with -( to start a barre and -) to end the barre.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-verbose` *marking-list* (list)

Make a fret diagram containing the symbols indicated in *marking-list*.

For example,

```
\markup \fret-diagram-verbose
#'( (mute 6) (mute 5) (open 4)
      (place-fret 3 2) (place-fret 2 3) (place-fret 1 2) )
```

produces a standard D chord diagram without fingering indications.

Possible elements in *marking-list*:

`(mute string-number)`

Place a small ,x‘ at the top of string *string-number*.

`(open string-number)`

Place a small ,o‘ at the top of string *string-number*.

`(barre start-string end-string fret-number)`

Place a barre indicator (much like a tie) from string *start-string* to string *end-string* at fret *fret-number*.

`(place-fret string-number fret-number finger-value)`

Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*. By default, the fret playing indicator is a solid dot. This can be changed by setting the value of the variable *dot-color*. If the *finger* part of the `place-fret` element is present, *finger-value* will be displayed according to the setting of the variable *finger-code*. There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)



`\harp-pedal` *definition-string* (string)

Make a harp pedal diagram.

Possible elements in *definition-string*:

- `^`            pedal is up
- `-`            pedal is neutral
- `v`            pedal is down
- `|`            vertical divider line
- `o`            the following pedal should be circled (indicating a change)

The function also checks if the string has the typical form of three pedals, then the divider and then the remaining four pedals. If not it prints out a warning. However, in any case, it will also print each symbol in the order as given. This means you can place the divider (even multiple dividers) anywhere you want, but you'll have to live with the warnings.

The appearance of the diagram can be tweaked *inter alia* using the `size` property of the `TextScript` grob (`\override Voice.TextScript #'size = #0.3`) for the overall, the `thickness` property (`\override Voice.TextScript #'thickness = #3`) for the line thickness of the horizontal line and the divider. The remaining configuration (box sizes, offsets and spaces) is done by the `harp-pedal-details` list of properties (`\override Voice.TextScript #'harp-pedal-details #'box-width = #1`). It contains the following settings: `box-offset` (vertical shift of the box center for up/down pedals), `box-width`, `box-height`, `space-before-divider` (the spacing between two boxes before the divider) and `space-after-divider` (box spacing after the divider).

`\markup \harp-pedal #"^~v|--ov^"`



Used properties:

- `thickness` (0.5)
- `harp-pedal-details`
- `size` (1.0)

### B.6.6 Other

`\backslashed-digit` *num* (integer)

A feta number, with backslash. This is for use in the context of figured bass notation.

```
\markup {
  \backslashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \backslashed-digit #7
}
```



Used properties:

- `thickness` (1.6)

- `font-size` (0)

`\char` *num* (integer)

Produce a single character. For example, `\char #65` produces the letter ,A‘.

```
\markup {
  \char #65
}
```

A

`\fraction` *arg1* (markup) *arg2* (markup)

Make a fraction of two markups.

```
\markup {
  \fraction 355 113
}
```

$$\pi \approx \frac{355}{113}$$

Used properties:

- `font-size` (0)

`\fromproperty` *symbol* (symbol)

Read the *symbol* from property settings, and produce a stencil from the markup contained within. If *symbol* is not defined, it returns an empty markup.

```
\header {
  myTitle = "myTitle"
  title = \markup {
    from
    \italic
    \fromproperty #'header:myTitle
  }
}
\markup {
  \null
}
```

**from *myTitle***

`\lookup` *glyph-name* (string)

Lookup a glyph by name.

```
\markup {
  \override #'(font-encoding . fetaBraces) {
    \lookup #"brace200"
    \hspace #2
    \rotate #180
    \lookup #"brace180"
  }
}
```

$$\left\{ \begin{array}{l} \left\{ \right. \\ \left. \right\} \end{array} \right\}$$

`\markalphabet` *num* (integer)

Make a markup letter for *num*. The letters start with A to Z and continue with double letters.

```
\markup {
  \markalphabet #8
  \hspace #2
  \markalphabet #26
}
```

I    AA

`\markletter` *num* (integer)

Make a markup letter for *num*. The letters start with A to Z (skipping letter I), and continue with double letters.

```
\markup {
  \markletter #8
  \hspace #2
  \markletter #26
}
```

J    AB

`\null`

An empty markup with extents of a single point.

```
\markup {
  \null
}
```

`\on-the-fly` *procedure* (symbol) *arg* (markup)

Apply the *procedure* markup command to *arg*. *procedure* should take a single argument.

`\override` *new-prop* (pair) *arg* (markup)

Add the first argument in to the property list. Properties may be any sort of property supported by [Abschnitt “font-interface” in Programmreferenz](#) and [Abschnitt “text-interface” in Programmreferenz](#), for example

```
\override #'(font-family . married) "bla"
\markup {
  \line {
    \column {
      default
      baseline-skip
    }
  }
  \hspace #2
  \override #'(baseline-skip . 4) {
    \column {
      increased
      baseline-skip
    }
  }
}
```

```

    }
  }
}

```

<b>default</b>	<b>increased</b>
<b>baseline-skip</b>	<b>baseline-skip</b>

`\page-ref` *label* (symbol) *gauge* (markup) *default* (markup)

Reference to a page number. *label* is the label set on the referenced page (using the `\label` command), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

`\slashed-digit` *num* (integer)

A feta number, with slash. This is for use in the context of figured bass notation.

```

\markup {
  \slashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \slashed-digit #7
}

```

**5 7**

Used properties:

- `thickness` (1.6)
- `font-size` (0)

`\stencil` *stil* (unknown)

Use a stencil as markup.

```

\markup {
  \stencil #(make-circle-stencil 2 0 #t)
}

```



`\strut`

Create a box of the same height as the space in the current font.

`\transparent` *arg* (markup)

Make the argument transparent.

```

\markup {
  \transparent {
    invisible text
  }
}

```

`\verbatim-file` *name* (string)

Read the contents of a file, and include it verbatim.

```

\markup {
  \verbatim-file #"simple.ly"
}

%% A simple piece in LilyPond, a scale.
\relative c' {
  c d e f g a b c
}
%% Optional helper for automatic updating by convert-ly. May be omitted.
\version "2.11.51"

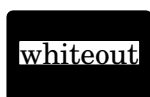
```

`\whiteout` *arg* (markup)  
Provide a white background for *arg*.

```

\markup {
  \combine
    \filled-box #'(-1 . 10) #'(-3 . 4) #1
    \whiteout whiteout
}

```



`\with-color` *color* (list) *arg* (markup)  
Draw *arg* in color specified by *color*.

```

\markup {
  \with-color #red
  red
  \hspace #2
  \with-color #green
  green
  \hspace #2
  \with-color #blue
  blue
}

```

red   green   blue

`\with-dimensions` *x* (pair of numbers) *y* (pair of numbers) *arg* (markup)  
Set the dimensions of *arg* to *x* and *y*.

## B.7 Text markup list commands

The following commands can all be used with `\markuplines`.

`\column-lines` *args* (list of markups)

Like `\column`, but return a list of lines instead of a single markup. `baseline-skip` determines the space between each markup in *args*.

Used properties:

- `baseline-skip`

`\justified-lines` *args* (list of markups)

Like `\justify`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\override-lines` *new-prop* (pair) *args* (list of markups)

Like `\override`, for markup lists.

`\wordwrap-internal` *justify* (boolean) *args* (list of markups)

Internal markup list command used to define `\justify` and `\wordwrap`.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)

`\wordwrap-lines` *args* (list of markups)

Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\wordwrap-string-internal` *justify* (boolean) *arg* (string)

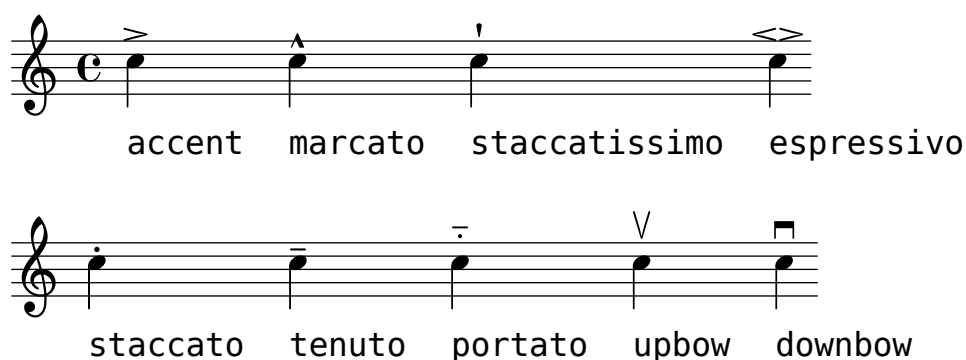
Internal markup list command used to define `\justify-string` and `\wordwrap-string`.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`

## B.8 Liste der Artikulationszeichen

Hier ist eine Liste, die alle möglichen Zeichen darstellt:



flageolet thumb lheel rheel ltoe rtoe

open stopped turn reverseturn trill prall

mordent prallprall prallmordent upprall

downprall upmordent downmordent pralldown

prallup lineprall signumcongruentiae shortfermata

fermata longfermata verylongfermata segno

coda varcoda

## B.9 Alle Kontexteigenschaften

`aDueText` (markup)

Text to print at a unisono passage.

`alignAboveContext` (string)

Where to insert newly created context in vertiical alignment.

`alignBassFigureAccidentals` (boolean)

If true, then the accidentals are aligned in bass figure context.

`alignBelowContext` (string)

Where to insert newly created context in vertiical alignment.

`associatedVoice` (string)

Name of the Voice that has the melody for this Lyrics line.

**autoAccidentals** (list)

List of different ways to typeset an accidental.

For determining when to print an accidental, several different rules are tried. The rule that gives the highest number of accidentals is used.

Each entry in the list is either a symbol or a procedure.

*symbol*      The symbol is the name of the context in which the following rules are to be applied. For example, if *context* is **Abschnitt “Score” in Programmreferenz** then all staves share accidentals, and if *context* is **Abschnitt “Staff” in Programmreferenz** then all voices in the same staff share accidentals, but staves do not.

*procedure*   The procedure represents an accidental rule to be applied to the previously specified context.

The procedure takes the following arguments:

**context**      The current context to which the rule should be applied.

**pitch**        The pitch of the note to be evaluated.

**barnum**       The current bar number.

**measurepos**      The current measure position.

The procedure returns a pair of booleans. The first states whether an extra natural should be added. The second states whether an accidental should be printed. (**#t** . **#f**) does not make sense.

**autoBeamCheck** (procedure)

A procedure taking three arguments, *context*, *dir* [start/stop (-1 or 1)], and *test* [shortest note in the beam]. A non-**#f** return value starts or stops the auto beam.

**autoBeamSettings** (list)

Specifies when automatically generated beams should begin and end. See **Abschnitt “Einstellung von automatischen Balken” in Benutzerhandbuch** for more information.

**autoBeaming** (boolean)

If set to true then beams are generated automatically.

**autoCautionaries** (list)

List similar to **autoAccidentals**, but it controls cautionary accidentals rather than normal ones. Both lists are tried, and the one giving the most accidentals wins. In case of draw, a normal accidental is typeset.

**automaticBars** (boolean)

If set to true then bar lines will not be printed automatically; they must be explicitly created with a **\bar** command. Unlike the **\cadenza** keyword, measures are still counted. Bar generation will resume according to that count if this property is unset.

**barAlways** (boolean)

If set to true a bar line is drawn after each note.

**barCheckSynchronize** (boolean)

If true then reset **measurePosition** when finding a bar check.

**barNumberVisibility** (procedure)

A Procedure that takes an integer and returns whether the corresponding bar number should be printed.



**bassFigureFormatFunction** (procedure)

A procedure that is called to produce the formatting for a **BassFigure** grob. It takes a list of **BassFigureEvents**, a context, and the grob to format.

**bassStaffProperties** (list)

An alist of property settings to apply for the down staff of **PianoStaff**. Used by `\autochange`.

**beatGrouping** (list)

A list of beatgroups, e.g., in 5/8 time '(2 3).

**beatLength** (moment)

The length of one beat in this time signature.

**chordChanges** (boolean)

Only show changes in chords scheme?

**chordNameExceptions** (list)

An alist of chord exceptions. Contains (*chord . markup*) entries.

**chordNameExceptionsFull** (list)

An alist of full chord exceptions. Contains (*chord . markup*) entries.

**chordNameExceptionsPartial** (list)

An alist of partial chord exceptions. Contains (*chord . (prefix-markup suffix-markup)*) entries.

**chordNameFunction** (procedure)

The function that converts lists of pitches to chord names.

**chordNameSeparator** (markup)

The markup object used to separate parts of a chord name.

**chordNoteNamer** (procedure)

A function that converts from a pitch object to a text markup. Used for single pitches.

**chordPrefixSpacer** (number)

The space added between the root symbol and the prefix of a chord name.

**chordRootNamer** (procedure)

A function that converts from a pitch object to a text markup. Used for chords.

**clefGlyph** (string)

Name of the symbol within the music font.

**clefOctavation** (integer)

Add this much extra octavation. Values of 7 and -7 are common.

**clefPosition** (number)

Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

**completionBusy** (boolean)

Whether a completion-note head is playing.

**connectArpeggios** (boolean)

If set, connect arpeggios across piano staff.

**countPercentRepeats** (boolean)

If set, produce counters for percent repeats.

`createKeyOnClefChange` (boolean)

Print a key signature whenever the clef is changed.

`createSpacing` (boolean)

Create `StaffSpacing` objects? Should be set for staves.

`crescendoSpanner` (symbol)

The type of spanner to be used for crescendi. Available values are ‘hairpin’, ‘line’, ‘dashed-line’, ‘dotted-line’. If unset, a hairpin crescendo is used.

`crescendoText` (markup)

The text to print at start of non-hairpin crescendo, i.e., ‘`cresc.`’.

`currentBarNumber` (integer)

Contains the current barnumber. This property is incremented at every bar line.

`decrescendoSpanner` (symbol)

See `crescendoSpanner`.

`decrescendoText` (markup)

The text to print at start of non-hairpin decrescendo, i.e., ‘`dim.`’.

`defaultBarType` (string)

Set the default type of bar line. See `whichBar` for information on available bar types.

This variable is read by [Abschnitt “Timing\\_translator” in Programmreferenz](#) at [Abschnitt “Score” in Programmreferenz](#) level.

`doubleSlurs` (boolean)

If set, two slurs are created for every slurred note, one above and one below the chord.

`drumPitchTable` (hash table)

A table mapping percussion instruments (symbols) to pitches.

`drumStyleTable` (hash table)

A hash table which maps drums to layout settings. Predefined values: ‘drums-style’, ‘timbales-style’, ‘congas-style’, ‘bongos-style’, and ‘percussion-style’.

The layout style is a hash table, containing the drum-pitches (e.g., the symbol ‘hihat’) as keys, and a list (*notehead-style script vertical-position*) as values.

`explicitClefVisibility` (vector)

‘break-visibility’ function for clef changes.

`explicitKeySignatureVisibility` (vector)

‘break-visibility’ function for explicit key changes. ‘\override’ of the `break-visibility` property will set the visibility for normal (i.e., at the start of the line) key signatures.

`extendersOverRests` (boolean)

Whether to continue extenders as they cross a rest.

`extraNatural` (boolean)

Whether to typeset an extra natural sign before accidentals changing from a non-natural to another non-natural.

`figuredBassAlterationDirection` (direction)

Where to put alterations relative to the main figure.

- figuredBassCenterContinuations** (boolean)  
Whether to vertically center pairs of extender lines. This does not work with three or more lines.
- figuredBassFormatter** (procedure)  
A routine generating a markup for a bass figure.
- figuredBassPlusDirection** (direction)  
Where to put plus signs relative to the main figure.
- fingeringOrientations** (list)  
A list of symbols, containing ‘left’, ‘right’, ‘up’ and/or ‘down’. This list determines where fingerings are put relative to the chord being fingered.
- firstClef** (boolean)  
If true, create a new clef when starting a staff.
- followVoice** (boolean)  
If set, note heads are tracked across staff switches by a thin line.
- fontSize** (number)  
The relative size of all grobs in a context.
- forbidBreak** (boolean)  
If set to **##t**, prevent a line break at this point.
- forceClef** (boolean)  
Show clef symbol, even if it has not changed. Only active for the first clef after the property is set, not for the full staff.
- gridInterval** (moment)  
Interval for which to generate **GridPoints**.
- harmonicAccidentals** (boolean)  
If set, harmonic notes in chords get accidentals.
- harmonicDots** (boolean)  
If set, harmonic notes in dotted chords get dots.
- highStringOne** (boolean)  
Whether the first string is the string with highest pitch on the instrument. This used by the automatic string selector for tablature notation.
- ignoreBarChecks** (boolean)  
Ignore bar checks.
- ignoreFiguredBassRest** (boolean)  
Don’t swallow rest events.
- ignoreMelismata** (boolean)  
Ignore melismata for this **Abschnitt “Lyrics” in Programmreferenz** line.
- implicitBassFigures** (list)  
A list of bass figures that are not printed as numbers, but only as extender lines.
- implicitTimeSignatureVisibility** (vector)  
break visibility for the default time signature.
- instrumentCueName** (markup)  
The name to print if another instrument is to be taken.
- instrumentEqualizer** (procedure)  
A function taking a string (instrument name), and returning a (*min* . *max*) pair of numbers for the loudness range of the instrument.

**instrumentName** (markup)

The name to print left of a staff. The **instrument** property labels the staff in the first system, and the **instr** property labels following lines.

**instrumentTransposition** (pitch)

Define the transposition of the instrument. Its value is the pitch that sounds like middle C. This is used to transpose the MIDI output, and `\quotes`.

**internalBarNumber** (integer)

Contains the current barnumber. This property is used for internal timekeeping, among others by the **Accidental\_engraver**.

**keepAliveInterfaces** (list)

A list of symbols, signifying grob interfaces that are worth keeping a staff with `remove-empty` set around for.

**keyAlterationOrder** (list)

An alist that defines in what order alterations should be printed. The format is `(step . alter)`, where *step* is a number from 0 to 6 and *alter* from -2 (sharp) to 2 (flat).

**keySignature** (list)

The current key signature. This is an alist containing `(step . alter)` or `((octave . step) . alter)`, where *step* is a number in the range 0 to 6 and *alter* a fraction, denoting alteration. For alterations, use symbols, e.g. `keySignature = #'((6 . ,FLAT))`.

**lyricMelismaAlignment** (direction)

Alignment to use for a melisma syllable.

**majorSevenSymbol** (markup)

How should the major 7th be formatted in a chord name?

**markFormatter** (procedure)

A procedure taking as arguments the context and the rehearsal mark. It should return the formatted mark as a markup object.

**maximumFretStretch** (number)

Don't allocate frets further than this from specified frets.

**measureLength** (moment)

Length of one measure in the current time signature.

**measurePosition** (moment)

How much of the current measure have we had. This can be set manually to create incomplete measures.

**melismaBusyProperties** (list)

A list of properties (symbols) to determine whether a melisma is playing. Setting this property will influence how lyrics are aligned to notes. For example, if set to  `#'(melismaBusy beamMelismaBusy)`, only manual melismata and manual beams are considered. Possible values include `melismaBusy`, `slurMelismaBusy`, `tieMelismaBusy`, and `beamMelismaBusy`.

**metronomeMarkFormatter** (procedure)

How to produce a metronome markup. Called with four arguments: text, duration, count and context.

**middleCClefPosition** (number)

The position of the middle C, as determined only by the clef. This can be calculated by looking at `clefPosition` and `clefGlyph`.

- middleCOffset** (number)  
The offset of middle C from the position given by **middleCClefPosition**. This is used for ottava brackets.
- middleCPosition** (number)  
The place of the middle C, measured in half staff-spaces. Usually determined by looking at **middleCClefPosition** and **middleCOffset**.
- midiInstrument** (string)  
Name of the MIDI instrument to use.
- midiMaximumVolume** (number)  
Analogous to **midiMinimumVolume**.
- midiMinimumVolume** (number)  
Set the minimum loudness for MIDI. Ranges from 0 to 1.
- minimumFret** (number)  
The tablature auto string-selecting mechanism selects the highest string with a fret at least **minimumFret**.
- minimumPageTurnLength** (moment)  
Minimum length of a rest for a page turn to be allowed.
- minimumRepeatLengthForPageTurn** (moment)  
Minimum length of a repeated section for a page turn to be allowed within that section.
- noteToFretFunction** (procedure)  
How to produce a fret diagram. Parameters: A list of note events and a list of tabstring events.
- ottavation** (markup)  
If set, the text for an ottava spanner. Changing this creates a new text spanner.
- output** (unknown)  
The output produced by a score-level translator during music interpretation.
- pedalSostenutoStrings** (list)  
See **pedalSustainStrings**.
- pedalSostenutoStyle** (symbol)  
See **pedalSustainStyle**.
- pedalSustainStrings** (list)  
A list of strings to print for sustain-pedal. Format is (*up updown down*), where each of the three is the string to print when this is done with the pedal.
- pedalSustainStyle** (symbol)  
A symbol that indicates how to print sustain pedals: **text**, **bracket** or **mixed** (both).
- pedalUnaCordaStrings** (list)  
See **pedalSustainStrings**.
- pedalUnaCordaStyle** (symbol)  
See **pedalSustainStyle**.
- predefinedDiagramTable** (hash table)  
The hash table of predefined fret diagrams to use in **FretBoards**.
- printKeyCancellation** (boolean)  
Print restoration alterations before a key signature change.

- printOctaveNames** (boolean)  
Print octave marks for the **NoteNames** context.
- printPartCombineTexts** (boolean)  
Set ‚Solo‘ and ‚A due‘ texts in the part combiner?
- proportionalNotationDuration** (moment)  
Global override for shortest-playing duration. This is used for switching on proportional notation.
- recordEventSequence** (procedure)  
When **Recording\_group\_engraver** is in this context, then upon termination of the context, this function is called with current context and a list of music objects. The list contains entries with start times, music objects and whether they are processed in this context.
- rehearsalMark** (integer)  
The last rehearsal mark printed.
- repeatCommands** (list)  
This property is a list of commands of the form (list 'volta x), where x is a string or #f. 'end-repeat is also accepted as a command.
- restNumberThreshold** (number)  
If a multimeasure rest has more measures than this, a number is printed.
- shapeNoteStyles** (vector)  
Vector of symbols, listing style for each note head relative to the tonic (qv.) of the scale.
- shortInstrumentName** (markup)  
See **instrument**.
- shortVocalName** (markup)  
Name of a vocal line, short version.
- skipBars** (boolean)  
If set to true, then skip the empty bars that are produced by multimeasure notes and rests. These bars will not appear on the printed output. If not set (the default), multimeasure notes and rests expand into their full length, printing the appropriate number of empty bars so that synchronization with other voices is preserved.
- ```
{
  r1 r1*3 R1*3
  \set Score.skipBars= ##t
  r1*3 R1*3
}
```
- skipTypesetting** (boolean)  
If true, no typesetting is done, speeding up the interpretation phase. Useful for debugging large scores.
- soloIIIText** (markup)  
The text for the start of a solo for voice ‚two‘ when part-combining.
- soloText** (markup)  
The text for the start of a solo when part-combining.
- squashedPosition** (integer)  
Vertical position of squashing for **Abschnitt „Pitch\_squash\_engraver“** in *Programmreferenz*.

**staffLineLayoutFunction** (procedure)

Layout of staff lines, **traditional**, or **semitone**.

**stanza** (markup)

Stanza ,number‘ to print before the start of a verse. Use in **Lyrics** context.

**stemLeftBeamCount** (integer)

Specify the number of beams to draw on the left side of the next note. Overrides automatic beaming. The value is only used once, and then it is erased.

**stemRightBeamCount** (integer)

See **stemLeftBeamCount**.

**stringNumberOrientations** (list)

See **fingeringOrientations**.

**stringOneTopmost** (boolean)

Whether the first string is printed on the top line of the tablature.

**stringTunings** (list)

The tablature strings tuning. It is a list of the pitch (in semitones) of each string (starting with the lower one).

**strokeFingerOrientations** (list)

See **fingeringOrientations**.

**subdivideBeams** (boolean)

If set, multiple beams will be subdivided at beat positions by only drawing one beam over the beat.

**suggestAccidentals** (boolean)

If set, accidentals are typeset as cautionary suggestions over the note.

**systemStartDelimiter** (symbol)

Which grob to make for the start of the system/staff? Set to **SystemStartBrace**, **SystemStartBracket** or **SystemStartBar**.

**systemStartDelimiterHierarchy** (pair)

A nested list, indicating the nesting of a start delimiters.

**tablatureFormat** (procedure)

A function formatting a tablature note head. Called with three arguments: string number, context and event. It returns the text as a string.

**tempoHideNote** (boolean)

Hide the note=count in tempo marks.

**tempoText** (markup)

Text for tempo marks.

**tempoUnitCount** (number)

Count for specifying tempo.

**tempoUnitDuration** (duration)

Unit for specifying tempo.

**tempoWholesPerMinute** (moment)

The tempo in whole notes per minute.

**tieWaitForNote** (boolean)

If true, tied notes do not have to follow each other directly. This can be used for writing out arpeggios.

**timeSignatureFraction** (pair of numbers)

A pair of numbers, signifying the time signature. For example, `#'(4 . 4)` is a 4/4 time signature.

**timing** (boolean)

Keep administration of measure length, position, bar number, etc.? Switch off for cadenzas.

**tonic** (pitch)

The tonic of the current scale.

**trebleStaffProperties** (list)

An alist of property settings to apply for the up staff of `PianoStaff`. Used by `\autochange`.

**tremoloFlags** (integer)

The number of tremolo flags to add if no number is specified.

**tupletFullLength** (boolean)

If set, the tuplet is printed up to the start of the next note.

**tupletFullLengthNote** (boolean)

If set, end at the next note, otherwise end on the matter (time signatures, etc.) before the note.

**tupletSpannerDuration** (moment)

Normally, a tuplet bracket is as wide as the `\times` expression that gave rise to it. By setting this property, you can make brackets last shorter.

```
{
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```

**useBassFigureExtenders** (boolean)

Whether to use extender lines for repeated bass figures.

**verticallySpacedContexts** (list)

List of symbols, containing context names whose vertical axis groups should be taken into account for vertical spacing of systems.

**vocalName** (markup)

Name of a vocal line.

**voltaSpannerDuration** (moment)

This specifies the maximum duration to use for the brackets printed for `\alternative`. This can be used to shrink the length of brackets in the situation where one alternative is very large.

**whichBar** (string)

This property is read to determine what type of bar line to create.

Example:

```
\set Staff.whichBar = "|: "
```

This will create a start-repeat bar in this staff only. Valid values are described in [Abschnitt “bar-line-interface” in \*Programmreferenz\*](#).



## B.10 Eigenschaften des Layouts

**X-extent** (pair of numbers)

Hard coded extent in X direction.

**X-offset** (number)

The horizontal amount that this object is moved relative to its X-parent.

**Y-extent** (pair of numbers)

Hard coded extent in Y direction.

**Y-offset** (number)

The vertical amount that this object is moved relative to its Y-parent.

**add-stem-support** (boolean)

If set, the **Stem** object is included in this script's support.

**after-line-breaking** (boolean)

Dummy property, used to trigger callback for **after-line-breaking**.

**align-dir** (direction)

Which side to align? -1: left side, 0: around center of width, 1: right side.

**allow-loose-spacing** (boolean)

If set, column can be detached from main spacing.

**allow-span-bar** (boolean)

If false, no inter-staff bar line will be created below this bar line.

**alteration** (number)

Alteration numbers for accidental.

**alteration-alist** (list)

List of (*pitch* . *accidental*) pairs for key signature.

**annotation** (string)

Annotate a grob for debug purposes.

**arpeggio-direction** (direction)

If set, put an arrow on the arpeggio squiggly line.

**arrow-length** (number)

Arrow length.

**arrow-width** (number)

Arrow width.

**auto-knee-gap** (dimension, in staff space)

If a gap is found between note heads where a horizontal beam fits that is larger than this number, make a kneed beam.

**average-spacing-wishes** (boolean)

If set, the spacing wishes are averaged over staves.

**avoid-note-head** (boolean)

If set, the stem of a chord does not pass through all note heads, but starts at the last note head.

**avoid-slur** (symbol)

Method of handling slur collisions. Choices are **around**, **inside**, **outside**. If unset, scripts and slurs ignore each other. **around** only moves the script if there is a collision; **outside** always moves the script.

- axes** (list) List of axis numbers. In the case of alignment grobs, this should contain only one number.
- bar-size** (dimension, in staff space)  
The size of a bar line.
- base-shortest-duration** (moment)  
Spacing is based on the shortest notes in a piece. Normally, pieces are spaced as if notes at least as short as this are present.
- baseline-skip** (dimension, in staff space)  
Distance between base lines of multiple lines of text.
- beam-thickness** (dimension, in staff space)  
Beam thickness, measured in **staff-space** units.
- beam-width** (dimension, in staff space)  
Width of the tremolo sign.
- beamed-stem-shorten** (list)  
How much to shorten beamed stems, when their direction is forced. It is a list, since the value is different depending on the number of flags and beams.
- beaming** (pair)  
Pair of number lists. Each number list specifies which beams to make. 0 is the central beam, 1 is the next beam toward the note, etc. This information is used to determine how to connect the beaming patterns from stem to stem inside a beam.
- before-line-breaking** (boolean)  
Dummy property, used to trigger a callback function.
- between-cols** (pair)  
Where to attach a loose column to.
- bound-details** (list)  
An alist of properties for determining attachments of spanners to edges.
- bound-padding** (number)  
The amount of padding to insert around spanner bounds.
- bracket-flare** (pair of numbers)  
A pair of numbers specifying how much edges of brackets should slant outward. Value 0.0 means straight edges.
- bracket-visibility** (boolean or symbol)  
This controls the visibility of the tuplet bracket. Setting it to false prevents printing of the bracket. Setting the property to **if-no-beam** makes it print only if there is no beam associated with this tuplet bracket.
- break-align-anchor** (number)  
Grobs aligned to this break-align grob will have their X-offsets shifted by this number. In bar lines, for example, this is used to position grobs relative to the (visual) center of the bar line.
- break-align-anchor-alignment** (number)  
Read by `ly:break-aligned-interface::calc-extent-aligned-anchor` for aligning an anchor to a grob's extent
- break-align-orders** (vector)  
Defines the order in which prefatory matter (clefs, key signatures) appears. The format is a vector of length 3, where each element is one order for end-of-line, middle of line, and start-of-line, respectively. An order is a list of symbols.

For example, clefs are put after key signatures by setting

```
\override Score.BreakAlignment #'break-align-orders =
  #(make-vector 3 '(span-bar
                    breathing-sign
                    staff-bar
                    key
                    clef
                    time-signature))
```

**break-align-symbol** (symbol)

This key is used for aligning and spacing breakable items.

**break-align-symbols** (list)

A list of symbols that determine which break-aligned grobs to align this to. If the grob selected by the first symbol in the list is invisible due to break-visibility, we will align to the next grob (and so on).

**break-overshoot** (pair of numbers)

How much does a broken spanner stick out of its bounds?

**break-visibility** (vector)

A vector of 3 booleans,  *#(end-of-line unbroken begin-of-line)*. *#t* means visible, *#f* means killed.

**breakable** (boolean)

Allow breaks here.

**c0-position** (integer)

An integer indicating the position of middle C.

**clip-edges** (boolean)

Allow outward pointing beamlets at the edges of beams?

**collapse-height** (dimension, in staff space)

Minimum height of system start delimiter. If equal or smaller, the bracket/brace/line is removed.

**color** (list)

The color of this grob.

**common-shortest-duration** (moment)

The most common shortest note length. This is used in spacing. Enlarging this sets the score tighter.

**concaveness** (number)

A beam is concave if its inner stems are closer to the beam than the two outside stems. This number is a measure of the closeness of the inner stems. It is used for damping the slope of the beam.

**connect-to-neighbor** (pair)

Pair of booleans, indicating whether this grob looks as a continued break.

**control-points** (list)

List of offsets (number pairs) that form control points for the tie, slur, or bracket shape. For Béziers, this should list the control points of a third-order Bézier curve.

**damping** (number)

Amount of beam slope damping.

- dash-fraction** (number)  
Size of the dashes, relative to **dash-period**. Should be between 0.0 (no line) and 1.0 (continuous line).
- dash-period** (number)  
The length of one dash together with whitespace. If negative, no line is drawn at all.
- default-direction** (direction)  
Direction determined by note head positions.
- digit-names** (unknown)  
Names for string finger digits.
- direction** (direction)  
If **side-axis** is 0 (or #X), then this property determines whether the object is placed #LEFT, #CENTER or #RIGHT with respect to the other object. Otherwise, it determines whether the object is placed #UP, #CENTER or #DOWN. Numerical values may also be used: #UP=1, #DOWN=-1, #LEFT=-1, #RIGHT=1, #CENTER=0.
- dot-count** (integer)  
The number of dots.
- dot-placement-list** (list)  
List consisting of (*description string-number fret-number finger-number*) entries used to define fret diagrams.
- duration-log** (integer)  
The 2-log of the note head duration, i.e., 0 = whole note, 1 = half note, etc.
- eccentricity** (number)  
How asymmetrical to make a slur. Positive means move the center to the right.
- edge-height** (pair)  
A pair of numbers specifying the heights of the vertical edges: (*left-height . right-height*).
- edge-text** (pair)  
A pair specifying the texts to be set at the edges: (*left-text . right-text*).
- expand-limit** (integer)  
Maximum number of measures expanded in church rests.
- extra-X-extent** (pair of numbers)  
A grob is enlarged in X dimension by this much.
- extra-Y-extent** (pair of numbers)  
A grob is enlarged in Y dimension by this much.
- extra-dy** (number)  
Slope glissandi this much extra.
- extra-offset** (pair of numbers)  
A pair representing an offset. This offset is added just before outputting the symbol, so the typesetting engine is completely oblivious to it. The values are measured in **staff-space** units of the staff's **StaffSymbol**.
- extra-spacing-height** (pair of numbers)  
In the horizontal spacing problem, we increase the height of each item by this amount (by adding the ,car' to the bottom of the item and adding the ,cdr' to the top of the item. In order to make a grob infinitely high (to prevent the horizontal spacing

problem from placing any other grobs above or below this grob), set this to `(-inf.0 . +inf.0)`.

**extra-spacing-width** (pair of numbers)

In the horizontal spacing problem, we pad each item by this amount (by adding the `,car‘` on the left side of the item and adding the `,cdr‘` on the right side of the item). In order to make a grob take up no horizontal space at all, set this to `(+inf.0 . -inf.0)`.

**flag** (unknown)

A function returning the full flag stencil for the **Stem**, which is passed to the function as the only argument. The default `ly:stem::calc-stencil` function uses the **flag-style** property to determine the correct glyph for the flag. By providing your own function, you can create arbitrary flags.

**flag-count** (number)

The number of tremolo beams.

**flag-style** (symbol)

A symbol determining what style of flag glyph is typeset on a **Stem**. Valid options include `'()` for standard flags, `'mensural` and `'no-flag`, which switches off the flag.

**font-encoding** (symbol)

The font encoding is the broadest category for selecting a font. Options include: `fetaMusic`, `fetaNumber`, `TeX-text`, `TeX-math`, `fetaBraces`, `fetaDynamic`.

**font-family** (symbol)

The font family is the broadest category for selecting text fonts. Options include: `sans`, `roman`.

**font-name** (string)

Specifies a file name (without extension) of the font to load. This setting overrides selection using **font-family**, **font-series** and **font-shape**.

**font-series** (symbol)

Select the series of a font. Choices include `medium`, `bold`, `bold-narrow`, etc.

**font-shape** (symbol)

Select the shape of a font. Choices include `upright`, `italic`, `caps`.

**font-size** (number)

The font size, compared to the `,normal‘` size. 0 is style-sheet’s normal size, `-1` is smaller, `+1` is bigger. Each step of 1 is approximately 12% larger; 6 steps are exactly a factor 2 larger. Fractional values are allowed.

**force-hshift** (number)

This specifies a manual shift for notes in collisions. The unit is the note head width of the first voice note. This is used by **Abschnitt “note-collision-interface” in Programmreferenz**.

**forced** (boolean)

Manually forced accidental.

**fraction** (pair of numbers)

Numerator and denominator of a time signature object.

**french-beaming** (boolean)

Use French beaming style for this stem. The stem stops at the innermost beams.

**fret-diagram-details** (list)

An alist of detailed grob properties for fret diagrams. Each alist entry consists of a (property . value) pair. The properties which can be included in fret-diagram-details include the following:

- **barre-type** – Type of barre indication used. Choices include **curved**, **straight**, and **none**. Default **curved**.
- **dot-color** – Color of dots. Options include **black** and **white**. Default **black**.
- **dot-label-font-mag** – Magnification for font used to label fret dots. Default value 1.
- **dot-radius** – Radius of dots, in terms of fret spaces. Default value 0.425 for labeled dots, 0.25 for unlabeled dots.
- **finger-code** – Code for the type of fingering indication used. Options include **none**, **in-dot**, and **below-string**. Default **none** for markup fret diagrams, **below-string** for FretBoards fret diagrams.
- **fret-count** – The number of frets. Default 4.
- **fret-label-font-mag** – The magnification of the font used to label the lowest fret number. Default 0.5
- **fret-label-vertical-offset** – The vertical offset of the fret label from the fret. Default -0.2
- **label-dir** – Side to which the fret label is attached. -1, **#LEFT**, or **#DOWN** for left or down; 1, **#RIGHT**, or **#UP** for right or up. Default **#RIGHT**.
- **mute-string** – Character string to be used to indicate muted string. Default "x".
- **number-type** – Type of numbers to use in fret label. Choices include **roman-lower**, **roman-upper**, and **arabic**. Default **roman-lower**.
- **open-string** – Character string to be used to indicate open string. Default "o".
- **orientation** – Orientation of fret-diagram. Options include **normal** and **landscape**. Default **normal**.
- **string-count** – The number of strings. Default 6.
- **string-label-font-mag** – The magnification of the font used to label fingerings at the string, rather than in the dot. Default value 0.6.
- **top-fret-thickness** – The thickness of the top fret line, as a multiple of the standard thickness. Default value 3.
- **xo-font-magnification** – Magnification used for mute and open string indicators. Default value 0.5.
- **xo-padding** – Padding for open and mute indicators from top fret. Default value 0.25.

**full-length-padding** (number)

How much padding to use at the right side of a full-length tuplet bracket.

**full-length-to-extent** (boolean)

Run to the extent of the column for a full-length tuplet bracket.

**full-size-change** (boolean)

Don't make a change clef smaller.

**gap** (dimension, in staff space)

Size of a gap in a variable symbol.

**gap-count** (integer)

Number of gapped beams for tremolo.

**glyph** (string)

A string determining what ‚style‘ of glyph is typeset. Valid choices depend on the function that is reading this property.

**glyph-name-alist** (list)

An alist of key-string pairs.

**grow-direction** (direction)

Crescendo or decrescendo?

**hair-thickness** (number)

Thickness of the thin line in a bar line.

**harp-pedal-details** (list)

An alist of detailed grob properties for harp pedal diagrams. Each alist entry consists of a (property . value) pair. The properties which can be included in harp-pedal-details include the following:

- **box-offset** – Vertical shift of the center of flat / sharp pedal boxes above / below the horizontal line. Default value 0.8.
- **box-width** – Width of each pedal box. Default value 0.4.
- **box-height** – Height of each pedal box. Default value 1.0.
- **space-before-divider** – Space between boxes before the first divider (so that the diagram can be made symmetric). Default value 0.8.
- **space-after-divider** – Space between boxes after the first divider. Default value 0.8.
- **circle-thickness** – Thickness (in unit of the line-thickness) of the ellipse around circled pedals. Default value 0.5.
- **circle-x-padding** – Padding in X direction of the ellipse around circled pedals. Default value 0.15.
- **circle-y-padding** – Padding in Y direction of the ellipse around circled pedals. Default value 0.2.

**head-direction** (direction)

Are the note heads left or right in a semitie?

**height** (dimension, in staff space)

Height of an object in **staff-space** units.

**height-limit** (dimension, in staff space)

Maximum slur height: The longer the slur, the closer it is to this height.

**horizontal-shift** (integer)

An integer that identifies ranking of **NoteColumns** for horizontal shifting. This is used by [Abschnitt “note-collision-interface” in Programmreferenz](#).

**horizontal-skylines** (unknown)

Two skylines, one to the left and one to the right of this grob.

**ignore-collision** (boolean)

If set, don’t do note collision resolution on this **NoteColumn**.

**implicit** (boolean)

Is this an implicit bass figure?

**inspect-index** (integer)

If debugging is set, set beam and slur configuration to this index, and print the respective scores.

**inspect-quants** (pair of numbers)

If debugging is set, set beam and slur quants to this position, and print the respective scores.

**keep-fixed-while-stretching** (boolean)

A grob with this property set to true is fixed relative to the staff above it when systems are stretched.

**keep-inside-line** (boolean)

If set, this column cannot have objects sticking into the margin.

**kern** (dimension, in staff space)

Amount of extra white space to add. For bar lines, this is the amount of space after a thick line.

**knee** (boolean)

Is this beam kneed?

**knee-spacing-correction** (number)

Factor for the optical correction amount for kneed beams. Set between 0 for no correction and 1 for full correction.

**labels** (list)

List of labels (symbols) placed on a column

**layer** (integer)

The output layer (a value between 0 and 2: Layers define the order of printing objects. Objects in lower layers are overprinted by objects in higher layers.

**ledger-line-thickness** (pair of numbers)

The thickness of ledger lines. It is the sum of 2 numbers: The first is the factor for line thickness, and the second for staff space. Both contributions are added.

**left-bound-info** (list)

An alist of properties for determining attachments of spanners to edges.

**left-padding** (dimension, in staff space)

The amount of space that is put left to an object (e.g., a group of accidentals).

**length** (dimension, in staff space)

User override for the stem length of unbeamed stems.

**length-fraction** (number)

Multiplier for lengths. Used for determining ledger lines and stem lengths.

**line-break-penalty** (number)

Penalty for a line break at this column. This affects the choices of the line breaker; it avoids a line break at a column with a positive penalty and prefers a line break at a column with a negative penalty.

**line-break-permission** (symbol)

Instructs the line breaker on whether to put a line break at this column. Can be **force** or **allow**.

**line-break-system-details** (list)

An alist of properties to use if this column is the start of a system.



- line-count** (integer)  
The number of staff lines.
- line-positions** (list)  
Vertical positions of staff lines.
- line-thickness** (number)  
The thickness of the tie or slur contour.
- long-text** (markup)  
Text markup. See [Abschnitt “Text formatieren” in Benutzerhandbuch](#).
- max-beam-connect** (integer)  
Maximum number of beams to connect to beams from this stem. Further beams are typeset as beamlets.
- max-stretch** (number)  
The maximum amount that this `VerticalAxisGroup` can be vertically stretched (for example, in order to better fill a page).
- measure-count** (integer)  
The number of measures for a multi-measure rest.
- measure-length** (moment)  
Length of a measure. Used in some spacing situations.
- merge-differently-dotted** (boolean)  
Merge note heads in collisions, even if they have a different number of dots. This is normal notation for some types of polyphonic music.  
**merge-differently-dotted** only applies to opposing stem directions (i.e., voice 1 & 2).
- merge-differently-headed** (boolean)  
Merge note heads in collisions, even if they have different note heads. The smaller of the two heads is rendered invisible. This is used in polyphonic guitar notation. The value of this setting is used by [Abschnitt “note-collision-interface” in Programmreferenz](#).  
**merge-differently-headed** only applies to opposing stem directions (i.e., voice 1 & 2).
- minimum-X-extent** (pair of numbers)  
Minimum size of an object in X dimension, measured in `staff-space` units.
- minimum-Y-extent** (pair of numbers)  
Minimum size of an object in Y dimension, measured in `staff-space` units.
- minimum-distance** (dimension, in staff space)  
Minimum distance between rest and notes or beam.
- minimum-length** (dimension, in staff space)  
Try to make a spanner at least this long, normally in the horizontal direction. This requires an appropriate callback for the `springs-and-rods` property. If added to a `Tie`, this sets the minimum distance between noteheads.
- minimum-length-fraction** (number)  
Minimum length of ledger line as fraction of note head size.
- minimum-space** (dimension, in staff space)  
Minimum distance that the victim should move (after padding).

- neutral-direction** (direction)  
Which direction to take in the center of the staff.
- neutral-position** (number)  
Position (in half staff spaces) where to flip the direction of custos stem.
- next** (layout object)  
Object that is next relation (e.g., the lyric syllable following an extender.
- no-alignment** (boolean)  
If set, don't place this grob in a **VerticalAlignment**; rather, place it using its own **Y-offset** callback.
- no-ledgers** (boolean)  
If set, don't draw ledger lines on this object.
- no-stem-extend** (boolean)  
If set, notes with ledger lines do not get stems extending to the middle staff line.
- non-default** (boolean)  
Set for manually specified clefs.
- non-musical** (boolean)  
True if the grob belongs to a **NonMusicalPaperColumn**.
- note-names** (vector)  
Vector of strings containing names for easy-notation note heads.
- outside-staff-horizontal-padding** (number)  
By default, an outside-staff-object can be placed so that is it very close to another grob horizontally. If this property is set, the outside-staff-object is raised so that it is not so close to its neighbor.
- outside-staff-padding** (number)  
The padding to place between this grob and the staff when spacing according to **outside-staff-priority**.
- outside-staff-priority** (number)  
If set, the grob is positioned outside the staff in such a way as to avoid all collisions. In case of a potential collision, the grob with the smaller **outside-staff-priority** is closer to the staff.
- packed-spacing** (boolean)  
If set, the notes are spaced as tightly as possible.
- padding** (dimension, in staff space)  
Add this much extra space between objects that are next to each other.
- page-break-penalty** (number)  
Penalty for page break at this column. This affects the choices of the page breaker; it avoids a page break at a column with a positive penalty and prefers a page break at a column with a negative penalty.
- page-break-permission** (symbol)  
Instructs the page breaker on whether to put a page break at this column. Can be **force** or **allow**.
- page-turn-penalty** (number)  
Penalty for a page turn at this column. This affects the choices of the page breaker; it avoids a page turn at a column with a positive penalty and prefers a page turn at a column with a negative penalty.

- page-turn-permission** (symbol)  
Instructs the page breaker on whether to put a page turn at this column. Can be **force** or **allow**.
- parenthesized** (boolean)  
Parenthesize this grob.
- positions** (pair of numbers)  
Pair of staff coordinates (*left* . *right*), where both *left* and *right* are in **staff-space** units of the current staff. For slurs, this value selects which slur candidate to use; if extreme positions are requested, the closest one is taken.
- prefer-dotted-right** (boolean)  
For note collisions, prefer to shift dotted up-note to the right, rather than shifting just the dot.
- ratio** (number)  
Parameter for slur shape. The higher this number, the quicker the slur attains its **height-limit**.
- remove-empty** (boolean)  
If set, remove group if it contains no interesting items.
- remove-first** (boolean)  
Remove the first staff of a orchestral score?
- restore-first** (boolean)  
Print a natural before the accidental.
- rhythmic-location** (rhythmic location)  
Where (bar number, measure position) in the score.
- right-bound-info** (list)  
An alist of properties for determining attachments of spanners to edges.
- right-padding** (dimension, in staff space)  
Space to insert on the right side of an object (e.g., between note and its accidentals).
- rotation** (list)  
Number of degrees to rotate this object, and what point to rotate around. For example, #'(45 0 0) rotates by 45 degrees around the center of this object.
- same-direction-correction** (number)  
Optical correction amount for stems that are placed in tight configurations. This amount is used for stems with the same direction to compensate for note head to stem distance.
- script-priority** (number)  
A sorting key that determines in what order a script is within a stack of scripts.
- self-alignment-X** (number)  
Specify alignment of an object. The value -1 means left aligned, 0 centered, and 1 right-aligned in X direction. Other numerical values may also be specified.
- self-alignment-Y** (number)  
Like **self-alignment-X** but for the Y axis.
- shorten-pair** (pair of numbers)  
The lengths to shorten a text-spanner on both sides, for example a pedal bracket. Positive values shorten the text-spanner, while negative values lengthen it.

**shortest-duration-space** (dimension, in staff space)

Start with this much space for the shortest duration. This is expressed in **spacing-increment** as unit. See also [Abschnitt “spacing-spanner-interface” in Programmreferenz](#).

**shortest-playing-duration** (moment)

The duration of the shortest note playing here.

**shortest-starter-duration** (moment)

The duration of the shortest note that starts here.

**side-axis** (number)

If the value is **#X** (or equivalently 0), the object is placed horizontally next to the other object. If the value is **#Y** or 1, it is placed vertically.

**side-relative-direction** (direction)

Multiply direction of **direction-source** with this to get the direction of this object.

**size** (number)

Size of object, relative to standard size.

**slope** (number)

The slope of this object.

**slur-padding** (number)

Extra distance between slur and script.

**space-alist** (list)

A table that specifies distances between prefatory items, like clef and time-signature. The format is an alist of spacing tuples: (*break-align-symbol type . distance*), where *type* can be the symbols **minimum-space** or **extra-space**.

**space-to-barline** (boolean)

If set, the distance between a note and the following non-musical column will be measured to the bar line instead of to the beginning of the non-musical column. If there is a clef change followed by a bar line, for example, this means that we will try to space the non-musical column as though the clef is not there.

**spacing-increment** (number)

Add this much space for a doubled duration. Typically, the width of a note head. See also [Abschnitt “spacing-spanner-interface” in Programmreferenz](#).

**springs-and-rods** (boolean)

Dummy variable for triggering spacing routines.

**stacking-dir** (direction)

Stack objects in which direction?

**staff-padding** (dimension, in staff space)

Maintain this much space between reference points and the staff. Its effect is to align objects of differing sizes (like the dynamics **p** and **f**) on their baselines.

**staff-position** (number)

Vertical position, measured in half staff spaces, counted from the middle line.

**staff-space** (dimension, in staff space)

Amount of space between staff lines, expressed in global **staff-space**.

**stem-attachment** (pair of numbers)

An (*x . y*) pair where the stem attaches to the notehead.

**stem-end-position** (number)

Where does the stem end (the end is opposite to the support-head)?

**stem-spacing-correction** (number)

Optical correction amount for stems that are placed in tight configurations. For opposite directions, this amount is the correction for two normal sized stems that overlap completely.

**stemlet-length** (number)

How long should a stem over a rest be?

**stencil** (unknown)

The symbol to print.

**stencils** (list)

Multiple stencils, used as intermediate value.

**strict-grace-spacing** (boolean)

If set, grace notes are not spaced separately, but put before musical columns.

**strict-note-spacing** (boolean)

If set, unbroken columns with non-musical material (clefs, bar lines, etc.) are not spaced separately, but put before musical columns.

**stroke-style** (string)

Set to "grace" to turn stroke through flag on.

**style** (symbol)

This setting determines in what style a grob is typeset. Valid choices depend on the **stencil** callback reading this property.

**text** (markup)

Text markup. See [Abschnitt "Text formatieren" in Benutzerhandbuch](#).

**text-direction** (direction)

This controls the ordering of the words. The default **RIGHT** is for roman text. Arabic or Hebrew should use **LEFT**.

**thick-thickness** (number)

Bar line thickness, measured in **line-thickness**.

**thickness** (number)

Line thickness, generally measured in **line-thickness**.

**thin-kern** (number)

The space after a hair-line in a bar line.

**threshold** (pair of numbers)

(*min* . *max*), where *min* and *max* are dimensions in staff space.

**tie-configuration** (list)

List of (*position* . *dir*) pairs, indicating the desired tie configuration, where *position* is the offset from the center of the staff in staff space and *dir* indicates the direction of the tie (1=>up, -1=>down, 0=>center). A non-pair entry in the list causes the corresponding tie to be formatted automatically.

**to-barline** (boolean)

If true, the spanner will stop at the bar line just before it would otherwise stop.

**transparent** (boolean)

This makes the grob invisible.

- uniform-stretching** (boolean)  
If set, items stretch proportionally to their durations. This looks better in complex polyphonic patterns.
- used** (boolean)  
If set, this spacing column is kept in the spacing problem.
- vertical-skylines** (unknown)  
Two skylines, one above and one below this grob.
- when** (moment)  
Global time step associated with this column happen?
- width** (dimension, in staff space)  
The width of a grob measured in staff space.
- word-space** (dimension, in staff space)  
Space to insert between words in texts.
- zigzag-length** (dimension, in staff space)  
The length of the lines of a zigzag, relative to **zigzag-width**. A value of 1 gives 60-degree zigzags.
- zigzag-width** (dimension, in staff space)  
The width of one zigzag squiggle. This number is adjusted slightly so that the glissando line can be constructed from a whole number of squiggles.

## B.11 Bezeichner

- acciaccatura** - *music* (music)  
(undocumented; fixme)
- addChordShape** - *key-symbol* (symbol) *shape-string* (string)  
(undocumented; fixme)
- addInstrumentDefinition** - *name* (string) *lst* (list)  
(undocumented; fixme)
- addQuote** - *name* (string) *music* (music)  
(undocumented; fixme)
- afterGrace** - *main* (music) *grace* (music)  
(undocumented; fixme)
- allowPageTurn**  
(undocumented; fixme)
- applyContext** - *proc* (procedure)  
(undocumented; fixme)
- applyMusic** - *func* (procedure) *music* (music)  
(undocumented; fixme)
- applyOutput** - *ctx* (symbol) *proc* (procedure)  
(undocumented; fixme)
- appoggiatura** - *music* (music)  
(undocumented; fixme)
- assertBeamQuant** - *l* (pair) *r* (pair)  
(undocumented; fixme)

**assertBeamSlope** - *comp* (procedure)  
 (undocumented; fixme)

**autochange** - *music* (music)  
 (undocumented; fixme)

**balloonGrobText** - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)  
 (undocumented; fixme)

**balloonText** - *offset* (pair of numbers) *text* (markup)  
 (undocumented; fixme)

**bar** - *type* (string)  
 (undocumented; fixme)

**barNumberCheck** - *n* (integer)  
 (undocumented; fixme)

**bendAfter** - *delta* (unknown)  
 (undocumented; fixme)

**breathe** (undocumented; fixme)

**clef** - *type* (string)  
 (undocumented; fixme)

**cueDuring** - *what* (string) *dir* (direction) *main-music* (music)  
 (undocumented; fixme)

**displayLilyMusic** - *music* (music)  
 (undocumented; fixme)

**displayMusic** - *music* (music)  
 (undocumented; fixme)

**endSpanners** - *music* (music)  
 (undocumented; fixme)

**featherDurations** - *factor* (moment) *argument* (music)  
 (undocumented; fixme)

**grace** - *music* (music)  
 (undocumented; fixme)

**includePageLayoutFile**  
 (undocumented; fixme)

**instrumentSwitch** - *name* (string)  
 (undocumented; fixme)

**keepWithTag** - *tag* (symbol) *music* (music)  
 (undocumented; fixme)

**killCues** - *music* (music)  
 (undocumented; fixme)

**label** - *label* (symbol)  
 (undocumented; fixme)

**makeClusters** - *arg* (music)  
 (undocumented; fixme)

**musicMap** - *proc* (procedure) *mus* (music)  
 (undocumented; fixme)

**noPageBreak**  
 (undocumented; fixme)

**noPageTurn**  
 (undocumented; fixme)

**octaveCheck** - *pitch-note* (music)  
 (undocumented; fixme)

**oldaddyrics** - *music* (music) *lyrics* (music)  
 (undocumented; fixme)

**ottava** - *octave* (number)  
 (undocumented; fixme)

**overrideProperty** - *name* (string) *property* (symbol) *value* (any type)  
 (undocumented; fixme)

**pageBreak**  
 (undocumented; fixme)

**pageTurn** (undocumented; fixme)

**parallelMusic** - *voice-ids* (list) *music* (music)  
 (undocumented; fixme)

**parenthesize** - *arg* (music)  
 (undocumented; fixme)

**partcombine** - *part1* (music) *part2* (music)  
 (undocumented; fixme)

**pitchedTrill** - *main-note* (music) *secondary-note* (music)  
 (undocumented; fixme)

**pointAndClickOff**  
 (undocumented; fixme)

**pointAndClickOn**  
 (undocumented; fixme)

**quoteDuring** - *what* (string) *main-music* (music)  
 (undocumented; fixme)

**removeWithTag** - *tag* (symbol) *music* (music)  
 (undocumented; fixme)

**resetRelativeOctave** - *reference-note* (music)  
 (undocumented; fixme)

**rightHandFinger** - *finger* (number or string)  
 (undocumented; fixme)

**scaleDurations** - *fraction* (pair of numbers) *music* (music)  
 (undocumented; fixme)

**scoreTweak** - *name* (string)  
 (undocumented; fixme)

**shiftDurations** - *dur* (integer) *dots* (integer) *arg* (music)  
 (undocumented; fixme)

**spacingTweaks** - *parameters* (list)  
 (undocumented; fixme)



**storePredefinedDiagram** - *chord* (music) *tuning* (list) *terse-definition* (string)  
 (undocumented; fixme)

**tag** - *tag* (symbol) *arg* (music)  
 (undocumented; fixme)

**tocItem** - *text* (markup)  
 Add a line to the table of content, using the **tocItemMarkup** paper variable markup

**transposedCueDuring** - *what* (string) *dir* (direction) *pitch-note* (music) *main-music* (music)  
 (undocumented; fixme)

**transposition** - *pitch-note* (music)  
 (undocumented; fixme)

**tweak** - *sym* (symbol) *val* (any type) *arg* (music)  
 (undocumented; fixme)

**unfoldRepeats** - *music* (music)  
 (undocumented; fixme)

**withMusicProperty** - *sym* (symbol) *val* (any type) *music* (music)  
 (undocumented; fixme)

## B.12 Scheme-Funktionen

**dispatcher** *x* [Funktion]  
 Is *x* a Dispatcher object?

**listener** *x* [Funktion]  
 Is *x* a Listener object?

**ly:add-file-name-alist** *alist* [Funktion]  
 Add mappings for error messages from *alist*.

**ly:add-interface** *a b c* [Funktion]  
 Add an interface description.

**ly:add-listener** *list disp cl* [Funktion]  
 Add the listener *list* to the dispatcher *disp*. Whenever *disp* hears an event of class *cl*, it is forwarded to *list*.

**ly:add-option** *sym val description* [Funktion]  
 Add a program option *sym* with default *val*.

**ly:all-grob-interfaces** [Funktion]  
 Get a hash table with all interface descriptions.

**ly:all-options** [Funktion]  
 Get all option settings in an alist.

**ly:all-stencil-expressions** [Funktion]  
 Return all symbols recognized as stencil expressions.

**ly:assoc-get** *key alist default-value* [Funktion]  
 Return value if *key* in *alist*, else **default-value** (or **#f** if not specified).

**ly:book-add-score!** *book-smob score* [Funktion]  
 Add *score* to *book-smob* score list.

- ly:book-process** *book-smob default-paper default-layout output* [Funktion]  
 Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).
- ly:book-process-to-systems** *book-smob default-paper default-layout output* [Funktion]  
 Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).
- ly:box?** *x* [Funktion]  
 Is *x* a Box object?
- ly:bp** *num* [Funktion]  
*num* bigpoints (1/72th inch).
- ly:bracket** *a iv t p* [Funktion]  
 Make a bracket in direction *a*. The extent of the bracket is given by *iv*. The wings protrude by an amount of *p*, which may be negative. The thickness is given by *t*.
- ly:broadcast** *disp ev* [Funktion]  
 Send the stream event *ev* to the dispatcher *disp*.
- ly:camel-case->lisp-identifier** *name-sym* [Funktion]  
 Convert FooBar\_Bla to foo-bar-bla style symbol.
- ly:chain-assoc-get** *key achain dfaul* [Funktion]  
 Return value for *key* from a list of alists *achain*. If no entry is found, return *dfa* or **#f** if no *dfa* is specified.
- ly:clear-anonymous-modules** [Funktion]  
 Plug a GUILE 1.6 and 1.7 memory leak by breaking a weak reference pointer cycle explicitly.
- ly:cm** *num* [Funktion]  
*num* cm.
- ly:command-line-code** [Funktion]  
 The Scheme code specified on command-line with ‘-e’.
- ly:command-line-options** [Funktion]  
 The Scheme options specified on command-line with ‘-d’.
- ly:command-line-verbose?** [Funktion]  
 Was *be\_verbose\_global* set?
- ly:connect-dispatchers** *to from* [Funktion]  
 Make the dispatcher *to* listen to events from *from*.
- ly:context-event-source** *context* [Funktion]  
 Return *event-source* of context *context*.
- ly:context-events-below** *context* [Funktion]  
 Return a *stream-distributor* that distributes all events from *context* and all its sub-contexts.
- ly:context-find** *context name* [Funktion]  
 Find a parent of *context* that has name or alias *name*. Return **#f** if not found.

|                                                                                                                                                                                                                     |            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>ly:context-grob-definition</b> <i>context name</i>                                                                                                                                                               | [Funktion] |
| Return the definition of <i>name</i> (a symbol) within <i>context</i> as an alist.                                                                                                                                  |            |
| <b>ly:context-id</b> <i>context</i>                                                                                                                                                                                 | [Funktion] |
| Return the ID string of <i>context</i> , i.e., for <code>\context Voice = one ...</code> return the string <code>one</code> .                                                                                       |            |
| <b>ly:context-name</b> <i>context</i>                                                                                                                                                                               | [Funktion] |
| Return the name of <i>context</i> , i.e., for <code>\context Voice = one ...</code> return the symbol <code>Voice</code> .                                                                                          |            |
| <b>ly:context-now</b> <i>context</i>                                                                                                                                                                                | [Funktion] |
| Return <code>now-moment</code> of context <i>context</i> .                                                                                                                                                          |            |
| <b>ly:context-parent</b> <i>context</i>                                                                                                                                                                             | [Funktion] |
| Return the parent of <i>context</i> , <code>#f</code> if none.                                                                                                                                                      |            |
| <b>ly:context-property</b> <i>c name</i>                                                                                                                                                                            | [Funktion] |
| Return the value of <i>name</i> from context <i>c</i> .                                                                                                                                                             |            |
| <b>ly:context-property-where-defined</b> <i>context name</i>                                                                                                                                                        | [Funktion] |
| Return the context above <i>context</i> where <i>name</i> is defined.                                                                                                                                               |            |
| <b>ly:context-pushpop-property</b> <i>context grob eltprop val</i>                                                                                                                                                  | [Funktion] |
| Do a single <code>\override</code> or <code>\revert</code> operation in <i>context</i> . The grob definition <i>grob</i> is extended with <i>eltprop</i> (if <i>val</i> is specified) or reverted (if unspecified). |            |
| <b>ly:context-set-property!</b> <i>context name val</i>                                                                                                                                                             | [Funktion] |
| Set value of property <i>name</i> in context <i>context</i> to <i>val</i> .                                                                                                                                         |            |
| <b>ly:context-unset-property</b> <i>context name</i>                                                                                                                                                                | [Funktion] |
| Unset value of property <i>name</i> in context <i>context</i> .                                                                                                                                                     |            |
| <b>ly:context?</b> <i>x</i>                                                                                                                                                                                         | [Funktion] |
| Is <i>x</i> a <code>Context</code> object?                                                                                                                                                                          |            |
| <b>ly:default-scale</b>                                                                                                                                                                                             | [Funktion] |
| Get the global default scale.                                                                                                                                                                                       |            |
| <b>ly:dimension?</b> <i>d</i>                                                                                                                                                                                       | [Funktion] |
| Return <i>d</i> as a number. Used to distinguish length variables from normal numbers.                                                                                                                              |            |
| <b>ly:dir?</b> <i>s</i>                                                                                                                                                                                             | [Funktion] |
| A type predicate. The direction <i>s</i> is <code>-1</code> , <code>0</code> or <code>1</code> , where <code>-1</code> represents left or down and <code>1</code> represents right or up.                           |            |
| <b>ly:duration-&gt;string</b> <i>dur</i>                                                                                                                                                                            | [Funktion] |
| Convert <i>dur</i> to a string.                                                                                                                                                                                     |            |
| <b>ly:duration-dot-count</b> <i>dur</i>                                                                                                                                                                             | [Funktion] |
| Extract the dot count from <i>dur</i> .                                                                                                                                                                             |            |
| <b>ly:duration-factor</b> <i>dur</i>                                                                                                                                                                                | [Funktion] |
| Extract the compression factor from <i>dur</i> . Return it as a pair.                                                                                                                                               |            |
| <b>ly:duration-length</b> <i>dur</i>                                                                                                                                                                                | [Funktion] |
| The length of the duration as a <code>moment</code> .                                                                                                                                                               |            |
| <b>ly:duration-log</b> <i>dur</i>                                                                                                                                                                                   | [Funktion] |
| Extract the duration log from <i>dur</i> .                                                                                                                                                                          |            |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <code>ly:duration&lt;? p1 p2</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | [Funktion] |
| Is <i>p1</i> shorter than <i>p2</i> ?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |
| <code>ly:duration? x</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | [Funktion] |
| Is <i>x</i> a <code>Duration</code> object?                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |            |
| <code>ly:effective-prefix</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | [Funktion] |
| Return effective prefix.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |            |
| <code>ly:error str rest</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | [Funktion] |
| A Scheme callable function to issue the error <i>str</i> . The error is formatted with <code>format</code> and <i>rest</i> .                                                                                                                                                                                                                                                                                                                                                                                            |            |
| <code>ly:eval-simple-closure delayed closure scm-start scm-end</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                   | [Funktion] |
| Evaluate a simple <i>closure</i> with the given <i>delayed</i> argument. If <i>scm-start</i> and <i>scm-end</i> are defined, evaluate it purely with those start and end points.                                                                                                                                                                                                                                                                                                                                        |            |
| <code>ly:event-deep-copy m</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | [Funktion] |
| Copy <i>m</i> and all sub expressions of <i>m</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |
| <code>ly:event-property sev sym</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | [Funktion] |
| Get the property <i>sym</i> of stream event <i>mus</i> . If <i>sym</i> is undefined, return '().                                                                                                                                                                                                                                                                                                                                                                                                                        |            |
| <code>ly:event-set-property! ev sym val</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | [Funktion] |
| Set property <i>sym</i> in event <i>ev</i> to <i>val</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                              |            |
| <code>ly:expand-environment str</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | [Funktion] |
| Expand <code>\$VAR</code> and <code>\${VAR}</code> in <i>str</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |
| <code>ly:export arg</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | [Funktion] |
| Export a Scheme object to the parser so it is treated as an identifier.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |            |
| <code>ly:find-accidentals-simple keysig pitch-scm barnum laziness octaveness</code>                                                                                                                                                                                                                                                                                                                                                                                                                                     | [Funktion] |
| Checks the need for an accidental and a 'restore' accidental against a key signature. The <i>laziness</i> is the number of bars for which reminder accidentals are used (ie. if <i>laziness</i> is zero, we only cancel accidentals in the same bar; if <i>laziness</i> is three, we cancel accidentals up to three bars after they first appear. <i>octaveness</i> is either ' <code>same-octave</code> ' or ' <code>any-octave</code> ' and it specifies whether accidentals should be canceled in different octaves. |            |
| <code>ly:find-file name</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | [Funktion] |
| Return the absolute file name of <i>name</i> , or <code>#f</code> if not found.                                                                                                                                                                                                                                                                                                                                                                                                                                         |            |
| <code>ly:font-config-display-fonts</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | [Funktion] |
| Dump a list of all fonts visible to <code>FontConfig</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                           |            |
| <code>ly:font-config-get-font-file name</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | [Funktion] |
| Get the file for font <i>name</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |
| <code>ly:font-design-size font</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | [Funktion] |
| Given the font metric <i>font</i> , return the design size, relative to the current output-scale.                                                                                                                                                                                                                                                                                                                                                                                                                       |            |
| <code>ly:font-file-name font</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | [Funktion] |
| Given the font metric <i>font</i> , return the corresponding file name.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |            |
| <code>ly:font-get-glyph font name</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | [Funktion] |
| Return a stencil from <i>font</i> for the glyph named <i>name</i> . If the glyph is not available, return an empty stencil.                                                                                                                                                                                                                                                                                                                                                                                             |            |

|                                                                                                                                                                |            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>ly:font-glyph-name-to-charcode</b> <i>font name</i>                                                                                                         | [Funktion] |
| Return the character code for glyph <i>name</i> in <i>font</i> .                                                                                               |            |
| <b>ly:font-glyph-name-to-index</b> <i>font name</i>                                                                                                            | [Funktion] |
| Return the index for <i>name</i> in <i>font</i> .                                                                                                              |            |
| <b>ly:font-index-to-charcode</b> <i>font index</i>                                                                                                             | [Funktion] |
| Return the character code for <i>index</i> in <i>font</i> .                                                                                                    |            |
| <b>ly:font-load</b> <i>name</i>                                                                                                                                | [Funktion] |
| Load the font <i>name</i> .                                                                                                                                    |            |
| <b>ly:font-magnification</b> <i>font</i>                                                                                                                       | [Funktion] |
| Given the font metric <i>font</i> , return the magnification, relative to the current output-scale.                                                            |            |
| <b>ly:font-metric?</b> <i>x</i>                                                                                                                                | [Funktion] |
| Is <i>x</i> a <code>Font_metric</code> object?                                                                                                                 |            |
| <b>ly:font-name</b> <i>font</i>                                                                                                                                | [Funktion] |
| Given the font metric <i>font</i> , return the corresponding name.                                                                                             |            |
| <b>ly:font-sub-fonts</b> <i>font</i>                                                                                                                           | [Funktion] |
| Given the font metric <i>font</i> of an OpenType font, return the names of the subfonts within <i>font</i> .                                                   |            |
| <b>ly:format</b> <i>str rest</i>                                                                                                                               | [Funktion] |
| LilyPond specific format, supporting <code>~a</code> and <code>~[0-9]f</code> .                                                                                |            |
| <b>ly:format-output</b> <i>context</i>                                                                                                                         | [Funktion] |
| Given a global context in its final state, process it and return the <code>Music_output</code> object in its final state.                                      |            |
| <b>ly:get-all-function-documentation</b>                                                                                                                       | [Funktion] |
| Get a hash table with all LilyPond Scheme extension functions.                                                                                                 |            |
| <b>ly:get-all-translators</b>                                                                                                                                  | [Funktion] |
| Return a list of all translator objects that may be instantiated.                                                                                              |            |
| <b>ly:get-glyph</b> <i>font index</i>                                                                                                                          | [Funktion] |
| Retrieve a stencil for the glyph numbered <i>index</i> in <i>font</i> .                                                                                        |            |
| <b>ly:get-listened-event-classes</b>                                                                                                                           | [Funktion] |
| Return a list of all event classes that some translator listens to.                                                                                            |            |
| <b>ly:get-option</b> <i>var</i>                                                                                                                                | [Funktion] |
| Get a global option setting.                                                                                                                                   |            |
| <b>ly:gettext</b> <i>original</i>                                                                                                                              | [Funktion] |
| A Scheme wrapper function for <code>gettext</code> .                                                                                                           |            |
| <b>ly:grob-alist-chain</b> <i>grob global</i>                                                                                                                  | [Funktion] |
| Get an alist chain for grob <i>grob</i> , with <i>global</i> as the global default. If unspecified, <code>font-defaults</code> from the layout block is taken. |            |
| <b>ly:grob-array-length</b> <i>grob-arr</i>                                                                                                                    | [Funktion] |
| Return the length of <i>grob-arr</i> .                                                                                                                         |            |

|                                                                                                                                                                                    |            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>ly:grob-array-ref</b> <i>grob-arr index</i>                                                                                                                                     | [Funktion] |
| Retrieve the <i>indexth</i> element of <i>grob-arr</i> .                                                                                                                           |            |
| <b>ly:grob-array?</b> <i>x</i>                                                                                                                                                     | [Funktion] |
| Is <i>x</i> a <i>Grob_array</i> object?                                                                                                                                            |            |
| <b>ly:grob-basic-properties</b> <i>grob</i>                                                                                                                                        | [Funktion] |
| Get the immutable properties of <i>grob</i> .                                                                                                                                      |            |
| <b>ly:grob-common-refpoint</b> <i>grob other axis</i>                                                                                                                              | [Funktion] |
| Find the common refpoint of <i>grob</i> and <i>other</i> for <i>axis</i> .                                                                                                         |            |
| <b>ly:grob-common-refpoint-of-array</b> <i>grob others axis</i>                                                                                                                    | [Funktion] |
| Find the common refpoint of <i>grob</i> and <i>others</i> (a <i>grob-array</i> ) for <i>axis</i> .                                                                                 |            |
| <b>ly:grob-default-font</b> <i>grob</i>                                                                                                                                            | [Funktion] |
| Return the default font for <i>grob</i> <i>gr</i> .                                                                                                                                |            |
| <b>ly:grob-extent</b> <i>grob refp axis</i>                                                                                                                                        | [Funktion] |
| Get the extent in <i>axis</i> direction of <i>grob</i> relative to the <i>grob refp</i> .                                                                                          |            |
| <b>ly:grob-interfaces</b> <i>grob</i>                                                                                                                                              | [Funktion] |
| Return the interfaces list of <i>grob</i> <i>grob</i> .                                                                                                                            |            |
| <b>ly:grob-layout</b> <i>grob</i>                                                                                                                                                  | [Funktion] |
| Get <i>\layout</i> definition from <i>grob</i> <i>grob</i> .                                                                                                                       |            |
| <b>ly:grob-object</b> <i>grob sym</i>                                                                                                                                              | [Funktion] |
| Return the value of a pointer in <i>grob</i> <i>g</i> of property <i>sym</i> . It returns '()' (end-of-list) if <i>sym</i> is undefined in <i>g</i> .                              |            |
| <b>ly:grob-original</b> <i>grob</i>                                                                                                                                                | [Funktion] |
| Return the unbroken original <i>grob</i> of <i>grob</i> .                                                                                                                          |            |
| <b>ly:grob-parent</b> <i>grob axis</i>                                                                                                                                             | [Funktion] |
| Get the parent of <i>grob</i> . <i>axis</i> is 0 for the X-axis, 1 for the Y-axis.                                                                                                 |            |
| <b>ly:grob-pq&lt;?</b> <i>a b</i>                                                                                                                                                  | [Funktion] |
| Compare two <i>grob</i> priority queue entries. This is an internal function.                                                                                                      |            |
| <b>ly:grob-properties</b> <i>grob</i>                                                                                                                                              | [Funktion] |
| Get the mutable properties of <i>grob</i> .                                                                                                                                        |            |
| <b>ly:grob-property</b> <i>grob sym deflt</i>                                                                                                                                      | [Funktion] |
| Return the value of a value in <i>grob</i> <i>g</i> of property <i>sym</i> . It returns '()' (end-of-list) or <i>deflt</i> (if specified) if <i>sym</i> is undefined in <i>g</i> . |            |
| <b>ly:grob-property-data</b> <i>grob sym</i>                                                                                                                                       | [Funktion] |
| Retrieve <i>sym</i> for <i>grob</i> but don't process callbacks.                                                                                                                   |            |
| <b>ly:grob-relative-coordinate</b> <i>grob refp axis</i>                                                                                                                           | [Funktion] |
| Get the coordinate in <i>axis</i> direction of <i>grob</i> relative to the <i>grob refp</i> .                                                                                      |            |
| <b>ly:grob-robust-relative-extent</b> <i>grob refp axis</i>                                                                                                                        | [Funktion] |
| Get the extent in <i>axis</i> direction of <i>grob</i> relative to the <i>grob refp</i> , or (0,0) if empty.                                                                       |            |
| <b>ly:grob-script-priority-less</b> <i>a b</i>                                                                                                                                     | [Funktion] |
| Compare two <i>grob</i> s by script priority. For internal use.                                                                                                                    |            |

|                                                                                                                                                                                 |            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>ly:grob-set-property!</b> <i>grob sym val</i>                                                                                                                                | [Funktion] |
| Set <i>sym</i> in grob <i>grob</i> to value <i>val</i> .                                                                                                                        |            |
| <b>ly:grob-staff-position</b> <i>sg</i>                                                                                                                                         | [Funktion] |
| Return the Y-position of <i>sg</i> relative to the staff.                                                                                                                       |            |
| <b>ly:grob-suicide!</b> <i>grob</i>                                                                                                                                             | [Funktion] |
| Kill <i>grob</i> .                                                                                                                                                              |            |
| <b>ly:grob-system</b> <i>grob</i>                                                                                                                                               | [Funktion] |
| Return the system grob of <i>grob</i> .                                                                                                                                         |            |
| <b>ly:grob-translate-axis!</b> <i>grob d a</i>                                                                                                                                  | [Funktion] |
| Translate <i>g</i> on axis <i>a</i> over distance <i>d</i> .                                                                                                                    |            |
| <b>ly:grob?</b> <i>x</i>                                                                                                                                                        | [Funktion] |
| Is <i>x</i> a Grob object?                                                                                                                                                      |            |
| <b>ly:gulp-file</b> <i>name size</i>                                                                                                                                            | [Funktion] |
| Read the file <i>name</i> , and return its contents in a string. The file is looked up using the search path.                                                                   |            |
| <b>ly:hash-table-keys</b> <i>tab</i>                                                                                                                                            | [Funktion] |
| Return a list of keys in <i>tab</i> .                                                                                                                                           |            |
| <b>ly:inch</b> <i>num</i>                                                                                                                                                       | [Funktion] |
| <i>num</i> inches.                                                                                                                                                              |            |
| <b>ly:input-both-locations</b> <i>sip</i>                                                                                                                                       | [Funktion] |
| Return input location in <i>sip</i> as (file-name first-line first-column last-line last-column).                                                                               |            |
| <b>ly:input-file-line-char-column</b> <i>sip</i>                                                                                                                                | [Funktion] |
| Return input location in <i>sip</i> as (file-name line char column).                                                                                                            |            |
| <b>ly:input-location?</b> <i>x</i>                                                                                                                                              | [Funktion] |
| Is <i>x</i> an input-location?                                                                                                                                                  |            |
| <b>ly:input-message</b> <i>sip msg rest</i>                                                                                                                                     | [Funktion] |
| Print <i>msg</i> as a GNU compliant error message, pointing to the location in <i>sip</i> . <i>msg</i> is interpreted similar to <b>format</b> 's argument, using <i>rest</i> . |            |
| <b>ly:interpret-music-expression</b> <i>mus ctx</i>                                                                                                                             | [Funktion] |
| Interpret the music expression <i>mus</i> in the global context <i>ctx</i> . The context is returned in its final state.                                                        |            |
| <b>ly:interpret-stencil-expression</b> <i>expr func arg1 offset</i>                                                                                                             | [Funktion] |
| Parse <i>expr</i> , feed bits to <i>func</i> with first arg <i>arg1</i> having offset <i>offset</i> .                                                                           |            |
| <b>ly:intlog2</b> <i>d</i>                                                                                                                                                      | [Funktion] |
| The 2-logarithm of 1/ <i>d</i> .                                                                                                                                                |            |
| <b>ly:is-listened-event-class</b> <i>sym</i>                                                                                                                                    | [Funktion] |
| Is <i>sym</i> a listened event class?                                                                                                                                           |            |
| <b>ly:item-break-dir</b> <i>it</i>                                                                                                                                              | [Funktion] |
| The break status direction of item <i>it</i> . -1 means end of line, 0 unbroken, and 1 beginning of line.                                                                       |            |

|                                                                                                                                                                                                                                                |            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <code>ly:item?</code> <i>g</i>                                                                                                                                                                                                                 | [Funktion] |
| Is <i>g</i> an <code>Item</code> object?                                                                                                                                                                                                       |            |
| <code>ly:iterator?</code> <i>x</i>                                                                                                                                                                                                             | [Funktion] |
| Is <i>x</i> a <code>Music_iterator</code> object?                                                                                                                                                                                              |            |
| <code>ly:lexer-keywords</code> <i>lexer</i>                                                                                                                                                                                                    | [Funktion] |
| Return a list of (KEY . CODE) pairs, signifying the LilyPond reserved words list.                                                                                                                                                              |            |
| <code>ly:lily-lexer?</code> <i>x</i>                                                                                                                                                                                                           | [Funktion] |
| Is <i>x</i> a <code>Lily_lexer</code> object?                                                                                                                                                                                                  |            |
| <code>ly:lily-parser?</code> <i>x</i>                                                                                                                                                                                                          | [Funktion] |
| Is <i>x</i> a <code>Lily_parser</code> object?                                                                                                                                                                                                 |            |
| <code>ly:load-text-dimensions</code> <i>dimension-alist</i>                                                                                                                                                                                    | [Funktion] |
| Load dimensions from T <sub>E</sub> X in a (KEY . (W H D)) format alist.                                                                                                                                                                       |            |
| <code>ly:make-book</code> <i>paper header scores</i>                                                                                                                                                                                           | [Funktion] |
| Make a <code>\book</code> of <i>paper</i> and <i>header</i> (which may be <code>#f</code> as well) containing <code>\scores</code> .                                                                                                           |            |
| <code>ly:make-dispatcher</code>                                                                                                                                                                                                                | [Funktion] |
| Return a newly created dispatcher.                                                                                                                                                                                                             |            |
| <code>ly:make-duration</code> <i>length dotcount num den</i>                                                                                                                                                                                   | [Funktion] |
| <i>length</i> is the negative logarithm (base 2) of the duration: 1 is a half note, 2 is a quarter note, 3 is an eighth note, etc. The number of dots after the note is given by the optional argument <i>dotcount</i> .                       |            |
| The duration factor is optionally given by <i>num</i> and <i>den</i> .                                                                                                                                                                         |            |
| A duration is a musical duration, i.e., a length of time described by a power of two (whole, half, quarter, etc.) and a number of augmentation dots.                                                                                           |            |
| <code>ly:make-global-context</code> <i>output-def</i>                                                                                                                                                                                          | [Funktion] |
| Set up a global interpretation context, using the output block <i>output-def</i> . The context is returned.                                                                                                                                    |            |
| <code>ly:make-global-translator</code> <i>global</i>                                                                                                                                                                                           | [Funktion] |
| Create a translator group and connect it to the global context <i>global</i> . The translator group is returned.                                                                                                                               |            |
| <code>ly:make-listener</code> <i>callback</i>                                                                                                                                                                                                  | [Funktion] |
| Create a listener. Any time the listener hears an object, it will call <i>callback</i> with that object. <i>callback</i> should take exactly one argument.                                                                                     |            |
| <code>ly:make-moment</code> <i>n d gn gd</i>                                                                                                                                                                                                   | [Funktion] |
| Create the rational number with main timing <i>n/d</i> , and optional grace timing <i>gn/gd</i> .                                                                                                                                              |            |
| A <i>moment</i> is a point in musical time. It consists of a pair of rationals ( <i>m</i> , <i>g</i> ), where <i>m</i> is the timing for the main notes, and <i>g</i> the timing for grace notes. In absence of grace notes, <i>g</i> is zero. |            |
| <code>ly:make-music</code> <i>props</i>                                                                                                                                                                                                        | [Funktion] |
| Make a C++ <code>Music</code> object and initialize it with <i>props</i> .                                                                                                                                                                     |            |
| This function is for internal use and is only called by <code>make-music</code> , which is the preferred interface for creating music objects.                                                                                                 |            |



- ly:make-music-function** *signature func* [Funktion]  
 Make a function to process music, to be used for the parser. **func** is the function, and **signature** describes its arguments. **signature** is a list containing either **ly:music?** predicates or other type predicates.
- ly:make-output-def** [Funktion]  
 Make an output definition.
- ly:make-page-label-marker** *label* [Funktion]  
 Return page marker with label.
- ly:make-page-permission-marker** *symbol permission* [Funktion]  
 Return page marker with page breaking and turning permissions.
- ly:make-pango-description-string** *chain size* [Funktion]  
 Make a PangoFontDescription string for the property alist *chain* at size *size*.
- ly:make-paper-outputter** *port format* [Funktion]  
 Create an outputter that evaluates within *output-format*, writing to *port*.
- ly:make-pitch** *octave note alter* [Funktion]  
*octave* is specified by an integer, zero for the octave containing middle C. *note* is a number from 0 to 6, with 0 corresponding to pitch C and 6 corresponding to pitch B. *alter* is a rational number of whole tones for alteration.
- ly:make-prob** *type init rest* [Funktion]  
 Create a Prob object.
- ly:make-scale** *steps* [Funktion]  
 Create a scale. The argument is a vector of rational numbers, each of which represents the number of tones of a pitch above the tonic.
- ly:make-score** *music* [Funktion]  
 Return score with *music* encapsulated in *score*.
- ly:make-simple-closure** *expr* [Funktion]  
 Make a simple closure. *expr* should be form of *(func a1 A2 ...)*, and will be invoked as *(func delayed-arg a1 a2 ...)*.
- ly:make-stencil** *expr xext yext* [Funktion]  
 Stencils are device independent output expressions. They carry two pieces of information:
1. A specification of how to print this object. This specification is processed by the output backends, for example 'scm/output-ps.scm'.
  2. The vertical and horizontal extents of the object, given as pairs. If an extent is unspecified (or if you use (1000 . -1000) as its value), it is taken to be empty.
- ly:make-stream-event** *cl proplist* [Funktion]  
 Create a stream event of class *cl* with the given mutable property list.
- ly:message** *str rest* [Funktion]  
 A Scheme callable function to issue the message *str*. The message is formatted with **format** and *rest*.
- ly:minimal-breaking** *pb* [Funktion]  
 Break (pages and lines) the **Paper\_book** object *pb* without looking for optimal spacing: stack as many lines on a page before moving to the next one.

|                                                                                                                                                                                                               |            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <code>ly:mm num</code><br><i>num</i> mm.                                                                                                                                                                      | [Funktion] |
| <code>ly:module-&gt;alist mod</code><br>Dump the contents of module <i>mod</i> as an alist.                                                                                                                   | [Funktion] |
| <code>ly:module-copy dest src</code><br>Copy all bindings from module <i>src</i> into <i>dest</i> .                                                                                                           | [Funktion] |
| <code>ly:modules-lookup modules sym def</code><br>Look up <i>sym</i> in the list <i>modules</i> , returning the first occurrence. If not found, return <i>def</i> or <i>#f</i> if <i>def</i> isn't specified. | [Funktion] |
| <code>ly:moment-add a b</code><br>Add two moments.                                                                                                                                                            | [Funktion] |
| <code>ly:moment-div a b</code><br>Divide two moments.                                                                                                                                                         | [Funktion] |
| <code>ly:moment-grace-denominator mom</code><br>Extract denominator from grace timing.                                                                                                                        | [Funktion] |
| <code>ly:moment-grace-numerator mom</code><br>Extract numerator from grace timing.                                                                                                                            | [Funktion] |
| <code>ly:moment-main-denominator mom</code><br>Extract denominator from main timing.                                                                                                                          | [Funktion] |
| <code>ly:moment-main-numerator mom</code><br>Extract numerator from main timing.                                                                                                                              | [Funktion] |
| <code>ly:moment-mod a b</code><br>Modulo of two moments.                                                                                                                                                      | [Funktion] |
| <code>ly:moment-mul a b</code><br>Multiply two moments.                                                                                                                                                       | [Funktion] |
| <code>ly:moment-sub a b</code><br>Subtract two moments.                                                                                                                                                       | [Funktion] |
| <code>ly:moment&lt;? a b</code><br>Compare two moments.                                                                                                                                                       | [Funktion] |
| <code>ly:moment? x</code><br>Is <i>x</i> a Moment object?                                                                                                                                                     | [Funktion] |
| <code>ly:music-compress m factor</code><br>Compress music object <i>m</i> by moment <i>factor</i> .                                                                                                           | [Funktion] |
| <code>ly:music-deep-copy m</code><br>Copy <i>m</i> and all sub expressions of <i>m</i> .                                                                                                                      | [Funktion] |
| <code>ly:music-duration-compress mus fact</code><br>Compress <i>mus</i> by factor <i>fact</i> , which is a Moment.                                                                                            | [Funktion] |
| <code>ly:music-duration-length mus</code><br>Extract the duration field from <i>mus</i> and return the length.                                                                                                | [Funktion] |

|                                                                                                                                                                                                |            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <code>ly:music-function-extract x</code>                                                                                                                                                       | [Funktion] |
| Return the Scheme function inside <i>x</i> .                                                                                                                                                   |            |
| <code>ly:music-function? x</code>                                                                                                                                                              | [Funktion] |
| Is <i>x</i> a <code>music-function</code> ?                                                                                                                                                    |            |
| <code>ly:music-length mus</code>                                                                                                                                                               | [Funktion] |
| Get the length of music expression <i>mus</i> and return it as a <code>Moment</code> object.                                                                                                   |            |
| <code>ly:music-list? lst</code>                                                                                                                                                                | [Funktion] |
| Type predicate: Return true if <i>lst</i> is a list of music objects.                                                                                                                          |            |
| <code>ly:music-mutable-properties mus</code>                                                                                                                                                   | [Funktion] |
| Return an alist containing the mutable properties of <i>mus</i> . The immutable properties are not available, since they are constant and initialized by the <code>make-music</code> function. |            |
| <code>ly:music-output? x</code>                                                                                                                                                                | [Funktion] |
| Is <i>x</i> a <code>Music_output</code> object?                                                                                                                                                |            |
| <code>ly:music-property mus sym dfault</code>                                                                                                                                                  | [Funktion] |
| Get the property <i>sym</i> of music expression <i>mus</i> . If <i>sym</i> is undefined, return '().                                                                                           |            |
| <code>ly:music-set-property! mus sym val</code>                                                                                                                                                | [Funktion] |
| Set property <i>sym</i> in music expression <i>mus</i> to <i>val</i> .                                                                                                                         |            |
| <code>ly:music-transpose m p</code>                                                                                                                                                            | [Funktion] |
| Transpose <i>m</i> such that central C is mapped to <i>p</i> . Return <i>m</i> .                                                                                                               |            |
| <code>ly:music? obj</code>                                                                                                                                                                     | [Funktion] |
| Type predicate.                                                                                                                                                                                |            |
| <code>ly:note-head::stem-attachment font-metric glyph-name</code>                                                                                                                              | [Funktion] |
| Get attachment in <i>font-metric</i> for attaching a stem to notehead <i>glyph-name</i> .                                                                                                      |            |
| <code>ly:number-&gt;string s</code>                                                                                                                                                            | [Funktion] |
| Convert <i>num</i> to a string without generating many decimals.                                                                                                                               |            |
| <code>ly:optimal-breaking pb</code>                                                                                                                                                            | [Funktion] |
| Optimally break (pages and lines) the <code>Paper_book</code> object <i>pb</i> to minimize badness in both vertical and horizontal spacing.                                                    |            |
| <code>ly:option-usage</code>                                                                                                                                                                   | [Funktion] |
| Print <code>ly:set-option</code> usage.                                                                                                                                                        |            |
| <code>ly:otf-&gt;cff otf-file-name</code>                                                                                                                                                      | [Funktion] |
| Convert the contents of an OTF file to a CFF file, returning it as a string.                                                                                                                   |            |
| <code>ly:otf-font-glyph-info font glyph</code>                                                                                                                                                 | [Funktion] |
| Given the font metric <i>font</i> of an OpenType font, return the information about named glyph <i>glyph</i> (a string).                                                                       |            |
| <code>ly:otf-font-table-data font tag</code>                                                                                                                                                   | [Funktion] |
| Extract a table <i>tag</i> from <i>font</i> . Return empty string for non-existent <i>tag</i> .                                                                                                |            |
| <code>ly:otf-font? font</code>                                                                                                                                                                 | [Funktion] |
| Is <i>font</i> an OpenType font?                                                                                                                                                               |            |

|                                                                                                                                                                         |            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <code>ly:otf-glyph-list</code> <i>font</i>                                                                                                                              | [Funktion] |
| Return a list of glyph names for <i>font</i> .                                                                                                                          |            |
| <code>ly:output-def-clone</code> <i>def</i>                                                                                                                             | [Funktion] |
| Clone output definition <i>def</i> .                                                                                                                                    |            |
| <code>ly:output-def-lookup</code> <i>pap sym def</i>                                                                                                                    | [Funktion] |
| Look up <i>sym</i> in the <i>pap</i> output definition (e.g., <code>\paper</code> ). Return the value or <i>def</i> (which defaults to <code>'()</code> ) if undefined. |            |
| <code>ly:output-def-parent</code> <i>def</i>                                                                                                                            | [Funktion] |
| Get the parent output definition of <i>def</i> .                                                                                                                        |            |
| <code>ly:output-def-scope</code> <i>def</i>                                                                                                                             | [Funktion] |
| Get the variable scope inside <i>def</i> .                                                                                                                              |            |
| <code>ly:output-def-set-variable!</code> <i>def sym val</i>                                                                                                             | [Funktion] |
| Set an output definition <i>def</i> variable <i>sym</i> to <i>val</i> .                                                                                                 |            |
| <code>ly:output-def?</code> <i>def</i>                                                                                                                                  | [Funktion] |
| Is <i>def</i> a layout definition?                                                                                                                                      |            |
| <code>ly:output-description</code> <i>output-def</i>                                                                                                                    | [Funktion] |
| Return the description of translators in <i>output-def</i> .                                                                                                            |            |
| <code>ly:output-formats</code>                                                                                                                                          | [Funktion] |
| Formats passed to <code>--format</code> as a list of strings, used for the output.                                                                                      |            |
| <code>ly:outputter-close</code> <i>outputter</i>                                                                                                                        | [Funktion] |
| Close port of <i>outputter</i> .                                                                                                                                        |            |
| <code>ly:outputter-dump-stencil</code> <i>outputter stencil</i>                                                                                                         | [Funktion] |
| Dump stencil <i>expr</i> onto <i>outputter</i> .                                                                                                                        |            |
| <code>ly:outputter-dump-string</code> <i>outputter str</i>                                                                                                              | [Funktion] |
| Dump <i>str</i> onto <i>outputter</i> .                                                                                                                                 |            |
| <code>ly:outputter-output-scheme</code> <i>outputter expr</i>                                                                                                           | [Funktion] |
| Eval <i>expr</i> in module of <i>outputter</i> .                                                                                                                        |            |
| <code>ly:outputter-port</code> <i>outputter</i>                                                                                                                         | [Funktion] |
| Return output port for <i>outputter</i> .                                                                                                                               |            |
| <code>ly:page-marker?</code> <i>x</i>                                                                                                                                   | [Funktion] |
| Is <i>x</i> a <code>Page_marker</code> object?                                                                                                                          |            |
| <code>ly:page-turn-breaking</code> <i>pb</i>                                                                                                                            | [Funktion] |
| Optimally break (pages and lines) the <code>Paper_book</code> object <i>pb</i> such that page turns only happen in specified places, returning its pages.               |            |
| <code>ly:pango-font-physical-fonts</code> <i>f</i>                                                                                                                      | [Funktion] |
| Return alist of (PSNAME . FILENAME) tuples.                                                                                                                             |            |
| <code>ly:pango-font?</code> <i>f</i>                                                                                                                                    | [Funktion] |
| Is <i>f</i> a pango font?                                                                                                                                               |            |
| <code>ly:paper-book-pages</code> <i>pb</i>                                                                                                                              | [Funktion] |
| Return pages in book <i>pb</i> .                                                                                                                                        |            |

|                                                                                                                                                         |            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <code>ly:paper-book-paper</code> <i>pb</i>                                                                                                              | [Funktion] |
| Return pages in book <i>pb</i> .                                                                                                                        |            |
| <code>ly:paper-book-performances</code> <i>paper-book</i>                                                                                               | [Funktion] |
| Return performances in book <i>paper-book</i> .                                                                                                         |            |
| <code>ly:paper-book-scopes</code> <i>book</i>                                                                                                           | [Funktion] |
| Return pages in layout book <i>book</i> .                                                                                                               |            |
| <code>ly:paper-book-systems</code> <i>pb</i>                                                                                                            | [Funktion] |
| Return systems in book <i>pb</i> .                                                                                                                      |            |
| <code>ly:paper-book?</code> <i>x</i>                                                                                                                    | [Funktion] |
| Is <i>x</i> a <code>Paper_book</code> object?                                                                                                           |            |
| <code>ly:paper-fonts</code> <i>bp</i>                                                                                                                   | [Funktion] |
| Return fonts from the <code>\paper</code> block <i>bp</i> .                                                                                             |            |
| <code>ly:paper-get-font</code> <i>paper-smob chain</i>                                                                                                  | [Funktion] |
| Return a font metric satisfying the font-qualifiers in the alist chain <i>chain</i> . (An alist chain is a list of alists, containing grob properties.) |            |
| <code>ly:paper-get-number</code> <i>layout-smob name</i>                                                                                                | [Funktion] |
| Return the layout variable <i>name</i> .                                                                                                                |            |
| <code>ly:paper-outputscales</code> <i>bp</i>                                                                                                            | [Funktion] |
| Get output-scale for <i>bp</i> .                                                                                                                        |            |
| <code>ly:paper-score-paper-systems</code> <i>paper-score</i>                                                                                            | [Funktion] |
| Return vector of <code>paper_system</code> objects from <i>paper-score</i> .                                                                            |            |
| <code>ly:paper-system-minimum-distance</code> <i>sys1 sys2</i>                                                                                          | [Funktion] |
| Measure the minimum distance between these two paper-systems, using their stored skylines if possible and falling back to their extents otherwise.      |            |
| <code>ly:paper-system?</code> <i>obj</i>                                                                                                                | [Funktion] |
| Type predicate.                                                                                                                                         |            |
| <code>ly:parse-file</code> <i>name</i>                                                                                                                  | [Funktion] |
| Parse a single <code>.ly</code> file. Upon failure, throw <code>ly-file-failed</code> key.                                                              |            |
| <code>ly:parser-clear-error</code> <i>parser</i>                                                                                                        | [Funktion] |
| Clear the error flag for the parser.                                                                                                                    |            |
| <code>ly:parser-clone</code> <i>parser-smob</i>                                                                                                         | [Funktion] |
| Return a clone of <i>parser-smob</i> .                                                                                                                  |            |
| <code>ly:parser-define!</code> <i>parser-smob symbol val</i>                                                                                            | [Funktion] |
| Bind <i>symbol</i> to <i>val</i> in <i>parser-smob</i> 's module.                                                                                       |            |
| <code>ly:parser-error</code> <i>parser msg input</i>                                                                                                    | [Funktion] |
| Display an error message and make the parser fail.                                                                                                      |            |
| <code>ly:parser-has-error?</code> <i>parser</i>                                                                                                         | [Funktion] |
| Does <i>parser</i> have an error flag?                                                                                                                  |            |
| <code>ly:parser-lexer</code> <i>parser-smob</i>                                                                                                         | [Funktion] |
| Return the lexer for <i>parser-smob</i> .                                                                                                               |            |

|                                                                                                                                       |            |
|---------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>ly:parser-lookup</b> <i>parser-smob symbol</i>                                                                                     | [Funktion] |
| Look up <i>symbol</i> in <i>parser-smob</i> 's module. Return '()' if not defined.                                                    |            |
| <b>ly:parser-output-name</b> <i>parser</i>                                                                                            | [Funktion] |
| Return the base name of the output file.                                                                                              |            |
| <b>ly:parser-parse-string</b> <i>parser-smob ly-code</i>                                                                              | [Funktion] |
| Parse the string <i>ly-code</i> with <i>parser-smob</i> . Upon failure, throw <b>ly-file-failed</b> key.                              |            |
| <b>ly:parser-set-note-names</b> <i>parser names</i>                                                                                   | [Funktion] |
| Replace current note names in <i>parser</i> . <i>names</i> is an alist of symbols. This only has effect if the current mode is notes. |            |
| <b>ly:performance-write</b> <i>performance filename</i>                                                                               | [Funktion] |
| Write <i>performance</i> to <i>filename</i> .                                                                                         |            |
| <b>ly:pfb-&gt;pfa</b> <i>pfb-file-name</i>                                                                                            | [Funktion] |
| Convert the contents of a PFB file to PFA.                                                                                            |            |
| <b>ly:pitch-alteration</b> <i>pp</i>                                                                                                  | [Funktion] |
| Extract the alteration from pitch <i>pp</i> .                                                                                         |            |
| <b>ly:pitch-diff</b> <i>pitch root</i>                                                                                                | [Funktion] |
| Return pitch <i>delta</i> such that <i>pitch</i> transposed by <i>delta</i> equals <i>root</i> .                                      |            |
| <b>ly:pitch-negate</b> <i>p</i>                                                                                                       | [Funktion] |
| Negate <i>p</i> .                                                                                                                     |            |
| <b>ly:pitch-notename</b> <i>pp</i>                                                                                                    | [Funktion] |
| Extract the note name from pitch <i>pp</i> .                                                                                          |            |
| <b>ly:pitch-octave</b> <i>pp</i>                                                                                                      | [Funktion] |
| Extract the octave from pitch <i>pp</i> .                                                                                             |            |
| <b>ly:pitch-quartertones</b> <i>pp</i>                                                                                                | [Funktion] |
| Calculate the number of quarter tones of <i>pp</i> from middle C.                                                                     |            |
| <b>ly:pitch-semitones</b> <i>pp</i>                                                                                                   | [Funktion] |
| Calculate the number of semitones of <i>pp</i> from middle C.                                                                         |            |
| <b>ly:pitch-steps</b> <i>p</i>                                                                                                        | [Funktion] |
| Number of steps counted from middle C of the pitch <i>p</i> .                                                                         |            |
| <b>ly:pitch-transpose</b> <i>p delta</i>                                                                                              | [Funktion] |
| Transpose <i>p</i> by the amount <i>delta</i> , where <i>delta</i> is relative to middle C.                                           |            |
| <b>ly:pitch&lt;?</b> <i>p1 p2</i>                                                                                                     | [Funktion] |
| Is <i>p1</i> lexicographically smaller than <i>p2</i> ?                                                                               |            |
| <b>ly:pitch?</b> <i>x</i>                                                                                                             | [Funktion] |
| Is <i>x</i> a Pitch object?                                                                                                           |            |
| <b>ly:position-on-line?</b> <i>sg spos</i>                                                                                            | [Funktion] |
| Return whether <i>pos</i> is on a line of the staff associated with the the grob <i>sg</i> (even on an extender line).                |            |
| <b>ly:prob-immutable-properties</b> <i>prob</i>                                                                                       | [Funktion] |
| Retrieve an alist of mutable properties.                                                                                              |            |

|                                                                                                                                                                             |            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>ly:prob-mutable-properties</b> <i>prob</i>                                                                                                                               | [Funktion] |
| Retrieve an alist of mutable properties.                                                                                                                                    |            |
| <b>ly:prob-property</b> <i>obj sym dfault</i>                                                                                                                               | [Funktion] |
| Return the value for <i>sym</i> .                                                                                                                                           |            |
| <b>ly:prob-property?</b> <i>obj sym</i>                                                                                                                                     | [Funktion] |
| Is boolean prop <i>sym</i> set?                                                                                                                                             |            |
| <b>ly:prob-set-property!</b> <i>obj sym value</i>                                                                                                                           | [Funktion] |
| Set property <i>sym</i> of <i>obj</i> to <i>value</i> .                                                                                                                     |            |
| <b>ly:prob-type?</b> <i>obj type</i>                                                                                                                                        | [Funktion] |
| Is <i>obj</i> the specified prob-type?                                                                                                                                      |            |
| <b>ly:prob?</b> <i>x</i>                                                                                                                                                    | [Funktion] |
| Is <i>x</i> a Prob object?                                                                                                                                                  |            |
| <b>ly:programming-error</b> <i>str rest</i>                                                                                                                                 | [Funktion] |
| A Scheme callable function to issue the internal warning <i>str</i> . The message is formatted with <b>format</b> and <i>rest</i> .                                         |            |
| <b>ly:progress</b> <i>str rest</i>                                                                                                                                          | [Funktion] |
| A Scheme callable function to print progress <i>str</i> . The message is formatted with <b>format</b> and <i>rest</i> .                                                     |            |
| <b>ly:property-lookup-stats</b> <i>sym</i>                                                                                                                                  | [Funktion] |
| Return hash table with a property access corresponding to <i>sym</i> . Choices are <b>prob</b> , <b>grob</b> , and <b>context</b> .                                         |            |
| <b>ly:protects</b>                                                                                                                                                          | [Funktion] |
| Return hash of protected objects.                                                                                                                                           |            |
| <b>ly:pt</b> <i>num</i>                                                                                                                                                     | [Funktion] |
| <i>num</i> printer points.                                                                                                                                                  |            |
| <b>ly:register-stencil-expression</b> <i>symbol</i>                                                                                                                         | [Funktion] |
| Add <i>symbol</i> as head of a stencil expression.                                                                                                                          |            |
| <b>ly:relative-group-extent</b> <i>elements common axis</i>                                                                                                                 | [Funktion] |
| Determine the extent of <i>elements</i> relative to <i>common</i> in the <i>axis</i> direction.                                                                             |            |
| <b>ly:reset-all-fonts</b>                                                                                                                                                   | [Funktion] |
| Forget all about previously loaded fonts.                                                                                                                                   |            |
| <b>ly:round-filled-box</b> <i>xext yext blot</i>                                                                                                                            | [Funktion] |
| Make a <b>Stencil</b> object that prints a black box of dimensions <i>xext</i> , <i>yext</i> and roundness <i>blot</i> .                                                    |            |
| <b>ly:round-filled-polygon</b> <i>points blot</i>                                                                                                                           | [Funktion] |
| Make a <b>Stencil</b> object that prints a black polygon with corners at the points defined by <i>points</i> (list of coordinate pairs) and roundness <i>blot</i> .         |            |
| <b>ly:run-translator</b> <i>mus output-def</i>                                                                                                                              | [Funktion] |
| Process <i>mus</i> according to <i>output-def</i> . An interpretation context is set up, and <i>mus</i> is interpreted with it. The context is returned in its final state. |            |
| Optionally, this routine takes an object-key to uniquely identify the score block containing it.                                                                            |            |

|                                                                                                                                                                                                                                                                                                                                                                                                                                            |            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>ly:score-add-output-def!</b> <i>score def</i>                                                                                                                                                                                                                                                                                                                                                                                           | [Funktion] |
| Add an output definition <i>def</i> to <i>score</i> .                                                                                                                                                                                                                                                                                                                                                                                      |            |
| <b>ly:score-embedded-format</b> <i>score layout</i>                                                                                                                                                                                                                                                                                                                                                                                        | [Funktion] |
| Run <i>score</i> through <i>layout</i> (an output definition) scaled to correct output-scale already, returning a list of layout-lines. This function takes an optional <b>Object_key</b> argument.                                                                                                                                                                                                                                        |            |
| <b>ly:score-error?</b> <i>score</i>                                                                                                                                                                                                                                                                                                                                                                                                        | [Funktion] |
| Was there an error in the score?                                                                                                                                                                                                                                                                                                                                                                                                           |            |
| <b>ly:score-header</b> <i>score</i>                                                                                                                                                                                                                                                                                                                                                                                                        | [Funktion] |
| Return score header.                                                                                                                                                                                                                                                                                                                                                                                                                       |            |
| <b>ly:score-music</b> <i>score</i>                                                                                                                                                                                                                                                                                                                                                                                                         | [Funktion] |
| Return score music.                                                                                                                                                                                                                                                                                                                                                                                                                        |            |
| <b>ly:score-output-defs</b> <i>score</i>                                                                                                                                                                                                                                                                                                                                                                                                   | [Funktion] |
| All output definitions in a score.                                                                                                                                                                                                                                                                                                                                                                                                         |            |
| <b>ly:score-set-header!</b> <i>score module</i>                                                                                                                                                                                                                                                                                                                                                                                            | [Funktion] |
| Set the score header.                                                                                                                                                                                                                                                                                                                                                                                                                      |            |
| <b>ly:score?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                                                                  | [Funktion] |
| Is <i>x</i> a <b>Score</b> object?                                                                                                                                                                                                                                                                                                                                                                                                         |            |
| <b>ly:set-default-scale</b> <i>scale</i>                                                                                                                                                                                                                                                                                                                                                                                                   | [Funktion] |
| Set the global default scale.                                                                                                                                                                                                                                                                                                                                                                                                              |            |
| <b>ly:set-grob-modification-callback</b> <i>cb</i>                                                                                                                                                                                                                                                                                                                                                                                         | [Funktion] |
| Specify a procedure that will be called every time LilyPond modifies a grob property. The callback will receive as arguments the grob that is being modified, the name of the C++ file in which the modification was requested, the line number in the C++ file in which the modification was requested, the name of the function in which the modification was requested, the property to be changed, and the new value for the property. |            |
| <b>ly:set-middle-C!</b> <i>context</i>                                                                                                                                                                                                                                                                                                                                                                                                     | [Funktion] |
| Set the <b>middleCPosition</b> variable in <i>context</i> based on the variables <b>middleCClefPosition</b> and <b>middleCOffset</b> .                                                                                                                                                                                                                                                                                                     |            |
| <b>ly:set-option</b> <i>var val</i>                                                                                                                                                                                                                                                                                                                                                                                                        | [Funktion] |
| Set a program option.                                                                                                                                                                                                                                                                                                                                                                                                                      |            |
| <b>ly:set-point-and-click</b> <i>what</i>                                                                                                                                                                                                                                                                                                                                                                                                  | [Funktion] |
| Deprecated.                                                                                                                                                                                                                                                                                                                                                                                                                                |            |
| <b>ly:set-property-cache-callback</b> <i>cb</i>                                                                                                                                                                                                                                                                                                                                                                                            | [Funktion] |
| Specify a procedure that will be called whenever lilypond calculates a callback function and caches the result. The callback will receive as arguments the grob whose property it is, the name of the property, the name of the callback that calculated the property, and the new (cached) value of the property.                                                                                                                         |            |
| <b>ly:simple-closure?</b> <i>clos</i>                                                                                                                                                                                                                                                                                                                                                                                                      | [Funktion] |
| Type predicate.                                                                                                                                                                                                                                                                                                                                                                                                                            |            |
| <b>ly:skyline-pair?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                                                           | [Funktion] |
| Is <i>x</i> a <b>Skyline_pair</b> object?                                                                                                                                                                                                                                                                                                                                                                                                  |            |



|                                                                                                                                                                                                                                                                                                                                                                                                                   |            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <code>ly:skyline? x</code>                                                                                                                                                                                                                                                                                                                                                                                        | [Funktion] |
| Is <i>x</i> a <i>Skyline</i> object?                                                                                                                                                                                                                                                                                                                                                                              |            |
| <code>ly:smob-protects</code>                                                                                                                                                                                                                                                                                                                                                                                     | [Funktion] |
| Return LilyPond's internal smob protection list.                                                                                                                                                                                                                                                                                                                                                                  |            |
| <code>ly:solve-spring-rod-problem springs rods length ragged</code>                                                                                                                                                                                                                                                                                                                                               | [Funktion] |
| Solve a spring and rod problem for <i>count</i> objects, that are connected by <i>count</i> -1 <i>springs</i> , and an arbitrary number of <i>rods</i> . <i>count</i> is implicitly given by <i>springs</i> and <i>rods</i> . The <i>springs</i> argument has the format ( <i>ideal</i> , <i>inverse_hook</i> ) and <i>rods</i> is of the form ( <i>idx1</i> , <i>idx2</i> , <i>distance</i> ).                   |            |
| <i>length</i> is a number, <i>ragged</i> a boolean.                                                                                                                                                                                                                                                                                                                                                               |            |
| The function returns a list containing the force (positive for stretching, negative for compressing and <i>#f</i> for non-satisfied constraints) followed by <i>spring-count</i> +1 positions of the objects.                                                                                                                                                                                                     |            |
| <code>ly:source-file? x</code>                                                                                                                                                                                                                                                                                                                                                                                    | [Funktion] |
| Is <i>x</i> a <i>Source_file</i> object?                                                                                                                                                                                                                                                                                                                                                                          |            |
| <code>ly:spanner-bound slur dir</code>                                                                                                                                                                                                                                                                                                                                                                            | [Funktion] |
| Get one of the bounds of <i>slur</i> . <i>dir</i> is -1 for left, and 1 for right.                                                                                                                                                                                                                                                                                                                                |            |
| <code>ly:spanner-broken-into spanner</code>                                                                                                                                                                                                                                                                                                                                                                       | [Funktion] |
| Return broken-into list for <i>spanner</i> .                                                                                                                                                                                                                                                                                                                                                                      |            |
| <code>ly:spanner? g</code>                                                                                                                                                                                                                                                                                                                                                                                        | [Funktion] |
| Is <i>g</i> a spanner object?                                                                                                                                                                                                                                                                                                                                                                                     |            |
| <code>ly:start-environment</code>                                                                                                                                                                                                                                                                                                                                                                                 | [Funktion] |
| Return the environment (a list of strings) that was in effect at program start.                                                                                                                                                                                                                                                                                                                                   |            |
| <code>ly:stderr-redirect file-name mode</code>                                                                                                                                                                                                                                                                                                                                                                    | [Funktion] |
| Redirect stderr to <i>file-name</i> , opened with <i>mode</i> .                                                                                                                                                                                                                                                                                                                                                   |            |
| <code>ly:stencil-add args</code>                                                                                                                                                                                                                                                                                                                                                                                  | [Funktion] |
| Combine stencils. Takes any number of arguments.                                                                                                                                                                                                                                                                                                                                                                  |            |
| <code>ly:stencil-aligned-to stil axis dir</code>                                                                                                                                                                                                                                                                                                                                                                  | [Funktion] |
| Align <i>stil</i> using its own extents. <i>dir</i> is a number. -1 and 1 are left and right, respectively. Other values are interpolated (so 0 means the center).                                                                                                                                                                                                                                                |            |
| <code>ly:stencil-combine-at-edge first axis direction second padding minimum</code>                                                                                                                                                                                                                                                                                                                               | [Funktion] |
| Construct a stencil by putting <i>second</i> next to <i>first</i> . <i>axis</i> can be 0 (x-axis) or 1 (y-axis). <i>direction</i> can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with <i>padding</i> as extra space. If this puts the reference points closer than <i>minimum</i> , they are moved by the latter amount. <i>first</i> and <i>second</i> may also be '()' or <i>#f</i> . |            |
| <code>ly:stencil-empty? stil</code>                                                                                                                                                                                                                                                                                                                                                                               | [Funktion] |
| Return whether <i>stil</i> is empty.                                                                                                                                                                                                                                                                                                                                                                              |            |
| <code>ly:stencil-expr stil</code>                                                                                                                                                                                                                                                                                                                                                                                 | [Funktion] |
| Return the expression of <i>stil</i> .                                                                                                                                                                                                                                                                                                                                                                            |            |
| <code>ly:stencil-extent stil axis</code>                                                                                                                                                                                                                                                                                                                                                                          | [Funktion] |
| Return a pair of numbers signifying the extent of <i>stil</i> in <i>axis</i> direction (0 or 1 for x and y axis, respectively).                                                                                                                                                                                                                                                                                   |            |

|                                                                                                                                                                                                                                                                                                                                                                                                                                              |            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>ly:stencil-fonts</b> <i>s</i>                                                                                                                                                                                                                                                                                                                                                                                                             | [Funktion] |
| Analyze <i>s</i> , and return a list of fonts used in <i>s</i> .                                                                                                                                                                                                                                                                                                                                                                             |            |
| <b>ly:stencil-in-color</b> <i>stc r g b</i>                                                                                                                                                                                                                                                                                                                                                                                                  | [Funktion] |
| Put <i>stc</i> in a different color.                                                                                                                                                                                                                                                                                                                                                                                                         |            |
| <b>ly:stencil-rotate</b> <i>stil angle x y</i>                                                                                                                                                                                                                                                                                                                                                                                               | [Funktion] |
| Return a stencil <i>stil</i> rotated <i>angle</i> degrees around point (x, y).                                                                                                                                                                                                                                                                                                                                                               |            |
| <b>ly:stencil-translate</b> <i>stil offset</i>                                                                                                                                                                                                                                                                                                                                                                                               | [Funktion] |
| Return a <i>stil</i> , but translated by <i>offset</i> (a pair of numbers).                                                                                                                                                                                                                                                                                                                                                                  |            |
| <b>ly:stencil-translate-axis</b> <i>stil amount axis</i>                                                                                                                                                                                                                                                                                                                                                                                     | [Funktion] |
| Return a copy of <i>stil</i> but translated by <i>amount</i> in <i>axis</i> direction.                                                                                                                                                                                                                                                                                                                                                       |            |
| <b>ly:stencil?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                                                                  | [Funktion] |
| Is <i>x</i> a Stencil object?                                                                                                                                                                                                                                                                                                                                                                                                                |            |
| <b>ly:stream-event?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                                                             | [Funktion] |
| Is <i>x</i> a Stream_event object?                                                                                                                                                                                                                                                                                                                                                                                                           |            |
| <b>ly:string-substitute</b> <i>a b s</i>                                                                                                                                                                                                                                                                                                                                                                                                     | [Funktion] |
| Replace string <i>a</i> by string <i>b</i> in string <i>s</i> .                                                                                                                                                                                                                                                                                                                                                                              |            |
| <b>ly:system-print</b> <i>system</i>                                                                                                                                                                                                                                                                                                                                                                                                         | [Funktion] |
| Draw the system and return the prob containing its stencil.                                                                                                                                                                                                                                                                                                                                                                                  |            |
| <b>ly:system-stretch</b> <i>system amount-scm</i>                                                                                                                                                                                                                                                                                                                                                                                            | [Funktion] |
| Stretch the system vertically by the given amount. This must be called before the system is drawn (for example with <b>ly:system-print</b> ).                                                                                                                                                                                                                                                                                                |            |
| <b>ly:text-dimension</b> <i>font text</i>                                                                                                                                                                                                                                                                                                                                                                                                    | [Funktion] |
| Given the font metric in <i>font</i> and the string <i>text</i> , compute the extents of that text in that font. The return value is a pair of number-pairs.                                                                                                                                                                                                                                                                                 |            |
| <b>ly:text-interface::interpret-markup</b>                                                                                                                                                                                                                                                                                                                                                                                                   | [Funktion] |
| Convert a text markup into a stencil. Takes three arguments, <i>layout</i> , <i>props</i> , and <i>markup</i> .<br><i>layout</i> is a \layout block; it may be obtained from a grob with <b>ly:grob-layout</b> . <i>props</i> is a alist chain, ie. a list of alists. This is typically obtained with ( <b>ly:grob-alist-chain</b> ( <b>ly:layout-lookup</b> layout 'text-font-defaults)). <i>markup</i> is the markup text to be processed. |            |
| <b>ly:translator-description</b> <i>me</i>                                                                                                                                                                                                                                                                                                                                                                                                   | [Funktion] |
| Return an alist of properties of translator <i>me</i> .                                                                                                                                                                                                                                                                                                                                                                                      |            |
| <b>ly:translator-group?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                                                         | [Funktion] |
| Is <i>x</i> a Translator_group object?                                                                                                                                                                                                                                                                                                                                                                                                       |            |
| <b>ly:translator-name</b> <i>trans</i>                                                                                                                                                                                                                                                                                                                                                                                                       | [Funktion] |
| Return the type name of the translator object <i>trans</i> . The name is a symbol.                                                                                                                                                                                                                                                                                                                                                           |            |
| <b>ly:translator?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                                                               | [Funktion] |
| Is <i>x</i> a Translator object?                                                                                                                                                                                                                                                                                                                                                                                                             |            |
| <b>ly:transpose-key-alist</b> <i>l pit</i>                                                                                                                                                                                                                                                                                                                                                                                                   | [Funktion] |
| Make a new key alist of <i>l</i> transposed by pitch <i>pit</i> .                                                                                                                                                                                                                                                                                                                                                                            |            |

|                                                                                                                            |            |
|----------------------------------------------------------------------------------------------------------------------------|------------|
| <code>ly:truncate-list!</code> <i>lst i</i>                                                                                | [Funktion] |
| Take at most the first <i>i</i> of list <i>lst</i> .                                                                       |            |
| <code>ly:ttf-&gt;pfa</code> <i>ttf-file-name</i>                                                                           | [Funktion] |
| Convert the contents of a TTF file to Type42 PFA, returning it as a string.                                                |            |
| <code>ly:ttf-ps-name</code> <i>ttf-file-name</i>                                                                           | [Funktion] |
| Extract the PostScript name from a TrueType font.                                                                          |            |
| <code>ly:unit</code>                                                                                                       | [Funktion] |
| Return the unit used for lengths as a string.                                                                              |            |
| <code>ly:usage</code>                                                                                                      | [Funktion] |
| Print usage message.                                                                                                       |            |
| <code>ly:version</code>                                                                                                    | [Funktion] |
| Return the current lilypond version as a list, e.g., (1 3 127 uu1).                                                        |            |
| <code>ly:warning</code> <i>str rest</i>                                                                                    | [Funktion] |
| A Scheme callable function to issue the warning <b>str</b> . The message is formatted with <b>format</b> and <b>rest</b> . |            |
| <code>ly:wide-char-&gt;utf-8</code> <i>wc</i>                                                                              | [Funktion] |
| Encode the Unicode codepoint <i>wc</i> , an integer, as UTF-8.                                                             |            |

## Anhang C Befehlsübersicht

### Syntax

1 2 8 16

### Erklärung

Tondauern

### Beispiel



c4. c4..

Punktierung



c d e f g a b

Tonleiter



fis bes

Vorzeichen



\clef treble \clef bass

Notenschlüssel



\time 3/4 \time 4/4

Taktangaben



r4 r8

Pause



d ~ d

Bindebogen



`\key es \major`

Tonart

`note'`

Oktavierung

`note,`

Oktavierung nach unten

`c( d e)`

Legatobogen

`c\ ( c( d) e\)`

Phrasierungsbogen

`a8[ b]`

Balken

`<< \new Staff ... >>`

mehr Notensysteme

`c-> c-.`

Artikulationszeichen



`c2\mf c\s fz`

Dynamik

`a\< a a\!`

Crescendo

`a\> a a\!`

Decrescendo

`< >`

Noten im Akkord

`\partial 8`

Auftakt

`\times 2/3 {f g a}`

Triolen

`\grace`

Verzierungen

`\lyricmode { twinkle }`

Texteingabe

twinkle

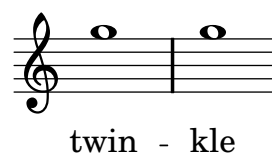
`\new Lyrics`

Textausgabe

twinkle

`twin -- kle`

Text-Trennstrich



```
\chordmode { c:dim f:maj7 }
```

Akkorde



```
\context ChordNames
```

Akkordsymbole drucken

 $C^{\circ} F^{\triangle}$ 

```
<<\{e f\} \\\{c d\}>>
```

Mehrstimmigkeit



```
s4 s8 s16
```

unsichtbare Pausen

# Anhang D GNU Free Documentation License

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ‘copyleft’, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The ‘Document’, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as ‘you’.

A ‘Modified Version’ of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ‘Secondary Section’ is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ‘Invariant Sections’ are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The ‘Cover Texts’ are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A ‘Transparent’ copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file



format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not ,Transparent‘ is called ,Opaque‘.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The ,Title Page‘ means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ,Title Page‘ means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled 'History', and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled 'History' in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the 'History' section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled 'Acknowledgments' or 'Dedications', preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled 'Endorsements'. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as 'Endorsements' or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to

the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled 'Endorsements', provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled 'History' in the various original documents, forming one section entitled 'History'; likewise combine any sections entitled 'Acknowledgments', and any sections entitled 'Dedications'. You must delete all sections entitled 'Endorsements'.

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an 'aggregate', and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License ‘or any later version’ applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## Anhang: Wie kann die Lizenz für eigene Dokumente verwendet werden

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
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 the Invariant Sections being list their titles, with the
Front-Cover Texts being list, and with the Back-Cover Texts being list.
A copy of the license is included in the section entitled 'GNU
Free Documentation License'
```

If you have no Invariant Sections, write `,with no Invariant Sections'` instead of saying which ones are invariant. If you have no Front-Cover Texts, write `,no Front-Cover Texts'` instead of `,Front-Cover Texts being list'`; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

## Anhang E Index der LilyPond-Befehle

Dieser Index listet alle LilyPond Befehle und Schlüsselwörter auf, versehen mit Verweisen zu den Abschnitten im Handbuch, die den Befehl beschreiben oder seine Verwendung diskutieren. Der erste Teil zeigt auf die genaue Stelle im Handbuch, an der der Befehl oder das Schlüsselwort erscheint, der zweite Teil zeigt auf den entsprechenden Abschnitt.

|                          |                                           |
|--------------------------|-------------------------------------------|
| !                        | ~                                         |
| ! ..... 5                | ~ ..... 244                               |
| ,                        | —                                         |
| ' ..... 1                | — ..... 174, 178                          |
| (                        | \                                         |
| (begin * * * *) ..... 57 | \! ..... 78                               |
| (end * * * *) ..... 57   | \( ..... 85                               |
| ,                        | \) ..... 85                               |
| ,                        | \< ..... 78                               |
| ,                        | \> ..... 78                               |
| ,                        | \abs-fontsize ..... 437                   |
| ,                        | \accepts ..... 361                        |
| ,                        | \addlyrics ..... 176                      |
| ,                        | \aeolian ..... 14                         |
| ,                        | \afterGrace ..... 70                      |
| ,                        | \aikenHeads ..... 27, 28                  |
| ,                        | \allowPageTurn ..... 323                  |
| ,                        | \alternative ..... 93                     |
| ,                        | \AncientRemoveEmptyStaffContext ..... 130 |
| ,                        | \applyContext ..... 409                   |
| ,                        | \applyOutput ..... 409                    |
| ,                        | \arpeggio ..... 88                        |
| ,                        | \arpeggioArrowDown ..... 88               |
| ,                        | \arpeggioArrowUp ..... 88                 |
| ,                        | \arpeggioBracket ..... 89                 |
| ,                        | \arpeggioNormal ..... 88                  |
| ,                        | \arpeggioParenthesis ..... 89             |
| ,                        | \arrow-head ..... 168, 460                |
| ,                        | \ascendens ..... 276                      |
| ,                        | \auctum ..... 276                         |
| ,                        | \augmentum ..... 276                      |
| ,                        | \autoBeamOff ..... 60                     |
| ,                        | \autoBeamOn ..... 60                      |
| ,                        | \autochange ..... 192                     |
| ,                        | \backslashed-digit ..... 470              |
| ,                        | \balloonGrobText ..... 149                |
| ,                        | \balloonLengthOff ..... 149               |
| ,                        | \balloonLengthOn ..... 149                |
| ,                        | \balloonText ..... 149                    |
| ,                        | \bar ..... 62                             |
| ,                        | \beam ..... 460                           |
| ,                        | \bendAfter ..... 86                       |
| ,                        | \bigger ..... 162, 438                    |
| ,                        | \bold ..... 162, 438                      |
| ,                        | \book ..... 287, 288                      |
| ,                        | \box ..... 167, 438                       |
| ,                        | \bracket ..... 82, 167, 460               |
| ,                        | \break ..... 321                          |
| ,                        | \breathe ..... 85                         |
| ,                        | \breve ..... 30, 38                       |
| ,                        | \cadenzaOff ..... 48                      |

|                                          |          |                                               |                    |
|------------------------------------------|----------|-----------------------------------------------|--------------------|
| <code>\cadenzaOn</code> .....            | 48       | <code>\fontsize</code> .....                  | 162, 439           |
| <code>\caesura</code> .....              | 269      | <code>\fp</code> .....                        | 78                 |
| <code>\caps</code> .....                 | 438      | <code>\fraction</code> .....                  | 471                |
| <code>\cavum</code> .....                | 276      | <code>\frenchChords</code> .....              | 249                |
| <code>\center-align</code> .....         | 164, 446 | <code>\fret-diagram</code> .....              | 468                |
| <code>\center-column</code> .....        | 165, 447 | <code>\fret-diagram-terse</code> .....        | 468                |
| <code>\change</code> .....               | 191      | <code>\fret-diagram-verbose</code> .....      | 469                |
| <code>\char</code> .....                 | 471      | <code>\fromproperty</code> .....              | 471                |
| <code>\chordmode</code> .....            | 4, 11    | <code>\general-align</code> .....             | 165, 449           |
| <code>\chords</code> .....               | 246      | <code>\germanChords</code> .....              | 249                |
| <code>\circle</code> .....               | 167, 461 | <code>\glissando</code> .....                 | 87                 |
| <code>\clef</code> .....                 | 11       | <code>\grace</code> .....                     | 70                 |
| <code>\column</code> .....               | 165, 447 | <code>\halign</code> .....                    | 164, 450           |
| <code>\column-lines</code> .....         | 474      | <code>\harp-pedal</code> .....                | 470                |
| <code>\combine</code> .....              | 168, 447 | <code>\hbracket</code> .....                  | 167, 462           |
| <code>\compressFullBarRests</code> ..... | 41       | <code>\hcenter-in</code> .....                | 451                |
| <code>\concat</code> .....               | 448      | <code>\header</code> .....                    | 288                |
| <code>\context</code> .....              | 357      | <code>\hideKeySignature</code> .....          | 238                |
| <code>\cr</code> .....                   | 78       | <code>\hideNotes</code> .....                 | 145                |
| <code>\crescHairpin</code> .....         | 78, 79   | <code>\hideStaffSwitch</code> .....           | 194                |
| <code>\crescTextCresc</code> .....       | 78, 79   | <code>\hspace</code> .....                    | 451                |
| <code>\decr</code> .....                 | 78       | <code>\huge</code> .....                      | 142, 143, 163, 440 |
| <code>\defaultTimeSignature</code> ..... | 45       | <code>\improvisationOff</code> .....          | 29                 |
| <code>\deminutum</code> .....            | 276      | <code>\improvisationOn</code> .....           | 29                 |
| <code>\denies</code> .....               | 361      | <code>\inclinatum</code> .....                | 276                |
| <code>\descendens</code> .....           | 276      | <code>\include</code> .....                   | 296                |
| <code>\dimHairpin</code> .....           | 78, 79   | <code>\ionian</code> .....                    | 14                 |
| <code>\dimTextDecr</code> .....          | 78, 79   | <code>\italianChords</code> .....             | 249                |
| <code>\dimTextDecresc</code> .....       | 78, 79   | <code>\italic</code> .....                    | 162, 440           |
| <code>\dimTextDim</code> .....           | 78, 79   | <code>\justified-lines</code> .....           | 475                |
| <code>\dir-column</code> .....           | 448      | <code>\justify</code> .....                   | 166, 452           |
| <code>\displayLilyMusic</code> .....     | 302, 402 | <code>\justify-field</code> .....             | 452                |
| <code>\displayMusic</code> .....         | 400      | <code>\justify-string</code> .....            | 453                |
| <code>\divisioMaior</code> .....         | 269      | <code>\keepWithTag</code> .....               | 299                |
| <code>\divisioMaxima</code> .....        | 269      | <code>\key</code> .....                       | 14, 27             |
| <code>\divisioMinima</code> .....        | 269      | <code>\label</code> .....                     | 294                |
| <code>\dorian</code> .....               | 14       | <code>\laissezVibrer</code> .....             | 36                 |
| <code>\dotsDown</code> .....             | 31       | <code>\large</code> .....                     | 142, 143, 163, 440 |
| <code>\dotsNeutral</code> .....          | 31       | <code>\larger</code> .....                    | 162, 440           |
| <code>\dotsUp</code> .....               | 31       | <code>\layout</code> .....                    | 288, 320           |
| <code>\doubleflat</code> .....           | 464      | <code>\left-align</code> .....                | 164, 453           |
| <code>\doublesharp</code> .....          | 464      | <code>\left-column</code> .....               | 454                |
| <code>\draw-circle</code> .....          | 168, 461 | <code>\line</code> .....                      | 454                |
| <code>\draw-line</code> .....            | 168, 461 | <code>\linea</code> .....                     | 276                |
| <code>\dynamic</code> .....              | 82, 439  | <code>\locrian</code> .....                   | 14                 |
| <code>\dynamicDown</code> .....          | 79       | <code>\longa</code> .....                     | 30, 38             |
| <code>\dynamicNeutral</code> .....       | 79       | <code>\lookup</code> .....                    | 471                |
| <code>\dynamicUp</code> .....            | 79       | <code>\lower</code> .....                     | 164, 454           |
| <code>\easyHeadsOff</code> .....         | 27       | <code>\lydian</code> .....                    | 14                 |
| <code>\easyHeadsOn</code> .....          | 27       | <code>\lyricmode</code> .....                 | 174, 176           |
| <code>\epsfile</code> .....              | 168, 461 | <code>\lyricsto</code> .....                  | 176                |
| <code>\espressivo</code> .....           | 78       | <code>\magnify</code> .....                   | 162, 440           |
| <code>\expandFullBarRests</code> .....   | 41       | <code>\major</code> .....                     | 14                 |
| <code>\f</code> .....                    | 78       | <code>\makeClusters</code> .....              | 102                |
| <code>\featherDurations</code> .....     | 61       | <code>\mark</code> .....                      | 68, 155            |
| <code>\ff</code> .....                   | 78       | <code>\markalphabet</code> .....              | 472                |
| <code>\fff</code> .....                  | 78       | <code>\markletter</code> .....                | 472                |
| <code>\ffff</code> .....                 | 78       | <code>\markup</code> .....                    | 159, 160           |
| <code>\fill-line</code> .....            | 166, 448 | <code>\markuplines</code> .....               | 159, 170           |
| <code>\filled-box</code> .....           | 168, 462 | <code>\maxima</code> .....                    | 30, 38             |
| <code>\finalis</code> .....              | 269      | <code>\medium</code> .....                    | 441                |
| <code>\finger</code> .....               | 143, 439 | <code>\melisma</code> .....                   | 180                |
| <code>\flat</code> .....                 | 465      | <code>\melismaEnd</code> .....                | 180                |
| <code>\flexa</code> .....                | 276      | <code>\mergeDifferentlyDottedOff</code> ..... | 106                |
| <code>\fontCaps</code> .....             | 439      | <code>\mergeDifferentlyDottedOn</code> .....  | 106                |



|                                                     |                    |                                       |                    |
|-----------------------------------------------------|--------------------|---------------------------------------|--------------------|
| <code>\mergeDifferentlyHeadedOff</code> .....       | 106                | <code>\repeatTie</code> .....         | 36, 94             |
| <code>\mergeDifferentlyHeadedOn</code> .....        | 106                | <code>\rest</code> .....              | 38                 |
| <code>\mf</code> .....                              | 78                 | <code>\rfz</code> .....               | 78                 |
| <code>\midi</code> .....                            | 288                | <code>\right-align</code> .....       | 164, 456           |
| <code>\minor</code> .....                           | 14                 | <code>\right-column</code> .....      | 456                |
| <code>\mixolydian</code> .....                      | 14                 | <code>\roman</code> .....             | 442                |
| <code>\mp</code> .....                              | 78                 | <code>\rotate</code> .....            | 457                |
| <code>\musicglyph</code> .....                      | 465                | <code>\rounded-box</code> .....       | 167, 463           |
| <code>\natural</code> .....                         | 465                | <code>\sacredHarpHeads</code> .....   | 27, 28             |
| <code>\new</code> .....                             | 357                | <code>\sans</code> .....              | 443                |
| <code>\noBeam</code> .....                          | 60                 | <code>\scaleDurations</code> .....    | 34                 |
| <code>\noBreak</code> .....                         | 321                | <code>\score</code> .....             | 286, 288, 466      |
| <code>\noPageBreak</code> .....                     | 322                | <code>\semiflat</code> .....          | 467                |
| <code>\noPageTurn</code> .....                      | 323                | <code>\semiGermanChords</code> .....  | 249                |
| <code>\normal-size-sub</code> .....                 | 441                | <code>\semisharp</code> .....         | 467                |
| <code>\normal-size-super</code> .....               | 441                | <code>\sesquiflat</code> .....        | 467                |
| <code>\normal-text</code> .....                     | 442                | <code>\sesquisharp</code> .....       | 467                |
| <code>\normalsize</code> .....                      | 142, 143, 163, 442 | <code>\set</code> .....               | 367                |
| <code>\note</code> .....                            | 465                | <code>\sf</code> .....                | 78                 |
| <code>\note-by-number</code> .....                  | 465                | <code>\sff</code> .....               | 78                 |
| <code>\null</code> .....                            | 472                | <code>\sfz</code> .....               | 78                 |
| <code>\number</code> .....                          | 442                | <code>\sharp</code> .....             | 467                |
| <code>\numericTimeSignature</code> .....            | 45                 | <code>\shiftOff</code> .....          | 106                |
| <code>\octaveCheck</code> .....                     | 7                  | <code>\shiftOn</code> .....           | 106                |
| <code>\on-the-fly</code> .....                      | 472                | <code>\shiftOnn</code> .....          | 106                |
| <code>\oneVoice</code> .....                        | 103                | <code>\shiftOnnn</code> .....         | 106                |
| <code>\oriscus</code> .....                         | 276                | <code>\showKeySignature</code> .....  | 238                |
| <code>\override</code> .....                        | 369, 472           | <code>\showStaffSwitch</code> .....   | 194                |
| <code>\override-lines</code> .....                  | 475                | <code>\simple</code> .....            | 443                |
| <code>\p</code> .....                               | 78                 | <code>\skip</code> .....              | 40                 |
| <code>\pad-around</code> .....                      | 167, 454           | <code>\slashed-digit</code> .....     | 473                |
| <code>\pad-markup</code> .....                      | 167, 455           | <code>\slurDashed</code> .....        | 84                 |
| <code>\pad-to-box</code> .....                      | 167, 455           | <code>\slurDotted</code> .....        | 84                 |
| <code>\pad-x</code> .....                           | 167, 455           | <code>\slurDown</code> .....          | 83, 84             |
| <code>\page-ref</code> .....                        | 294, 473           | <code>\slurNeutral</code> .....       | 83, 84             |
| <code>\pageBreak</code> .....                       | 322                | <code>\slurSolid</code> .....         | 84                 |
| <code>\pageTurn</code> .....                        | 323                | <code>\slurUp</code> .....            | 84                 |
| <code>\paper</code> .....                           | 288, 314           | <code>\small</code> .....             | 142, 143, 163, 443 |
| <code>\parallelMusic</code> .....                   | 113                | <code>\smallCaps</code> .....         | 443                |
| <code>\parenthesize</code> .....                    | 147                | <code>\smaller</code> .....           | 162, 444           |
| <code>\partcombine</code> .....                     | 109                | <code>\sostenutoOff</code> .....      | 196                |
| <code>\partial</code> .....                         | 47                 | <code>\sostenutoOn</code> .....       | 196                |
| <code>\partial</code> .....                         | 93                 | <code>\sp</code> .....                | 78                 |
| <code>\pes</code> .....                             | 276                | <code>\spp</code> .....               | 78                 |
| <code>\phrasingSlurDown</code> .....                | 85                 | <code>\startGroup</code> .....        | 152                |
| <code>\phrasingSlurNeutral</code> .....             | 85                 | <code>\startStaff</code> .....        | 124                |
| <code>\phrasingSlurUp</code> .....                  | 85                 | <code>\startTrillSpan</code> .....    | 91                 |
| <code>\phrygian</code> .....                        | 14                 | <code>\stemDown</code> .....          | 148                |
| <code>\pitchedTrill</code> .....                    | 91                 | <code>\stemNeutral</code> .....       | 148                |
| <code>\postscript</code> .....                      | 168, 462           | <code>\stemUp</code> .....            | 148                |
| <code>\pp</code> .....                              | 78                 | <code>\stencil</code> .....           | 473                |
| <code>\ppp</code> .....                             | 78                 | <code>\stopGroup</code> .....         | 152                |
| <code>\pppp</code> .....                            | 78                 | <code>\stopStaff</code> .....         | 124                |
| <code>\ppppp</code> .....                           | 78                 | <code>\stopTrillSpan</code> .....     | 91                 |
| <code>\property in \lyricmode</code> .....          | 174                | <code>\strophæ</code> .....           | 276                |
| <code>\put-adjacent</code> .....                    | 456                | <code>\strut</code> .....             | 473                |
| <code>\quilisma</code> .....                        | 276                | <code>\sub</code> .....               | 162, 444           |
| <code>\raise</code> .....                           | 164, 456           | <code>\super</code> .....             | 162, 444           |
| <code>\relative</code> .....                        | 2, 4, 11, 193      | <code>\sustainOff</code> .....        | 196                |
| <code>\RemoveEmptyRhythmicStaffContext</code> ..... | 130                | <code>\sustainOn</code> .....         | 196                |
| <code>\RemoveEmptyStaffContext</code> .....         | 129                | <code>\table-of-contents</code> ..... | 296                |
| <code>\removeWithTag</code> .....                   | 299                | <code>\tag</code> .....               | 299                |
| <code>\repeat</code> .....                          | 93                 | <code>\taor</code> .....              | 238                |
| <code>\repeat percent</code> .....                  | 99                 | <code>\teeny</code> .....             | 142, 143, 163, 445 |
| <code>\repeat tremolo</code> .....                  | 100                | <code>\tempo</code> .....             | 132                |



|                                              |                    |
|----------------------------------------------|--------------------|
| <code>\text</code> .....                     | 445                |
| <code>\textLengthOff</code> .....            | 154                |
| <code>\textLengthOn</code> .....             | 154                |
| <code>\thumb</code> .....                    | 144                |
| <code>\tied-lyric</code> .....               | 468                |
| <code>\tieDashed</code> .....                | 36                 |
| <code>\tieDotted</code> .....                | 36                 |
| <code>\tieDown</code> .....                  | 36                 |
| <code>\tieNeutral</code> .....               | 36                 |
| <code>\tieSolid</code> .....                 | 36                 |
| <code>\tieUp</code> .....                    | 36                 |
| <code>\time</code> .....                     | 44                 |
| <code>\times</code> .....                    | 32                 |
| <code>\tiny</code> .....                     | 142, 143, 163, 445 |
| <code>\tocItem</code> .....                  | 296                |
| <code>\translate</code> .....                | 165, 457           |
| <code>\translate-scaled</code> .....         | 165, 457           |
| <code>\transparent</code> .....              | 473                |
| <code>\transpose</code> .....                | 4, 8, 11           |
| <code>\transposition</code> .....            | 17                 |
| <code>\treCorde</code> .....                 | 196                |
| <code>\triangle</code> .....                 | 168, 464           |
| <code>\trill</code> .....                    | 91                 |
| <code>\tupletDown</code> .....               | 32                 |
| <code>\tupletNeutral</code> .....            | 32                 |
| <code>\tupletUp</code> .....                 | 32                 |
| <code>\tweak</code> .....                    | 369                |
| <code>\typewriter</code> .....               | 445                |
| <code>\unaCorda</code> .....                 | 196                |
| <code>\underline</code> .....                | 162, 446           |
| <code>\unfoldRepeats</code> .....            | 307                |
| <code>\unHideNotes</code> .....              | 145                |
| <code>\unset</code> .....                    | 368                |
| <code>\upright</code> .....                  | 446                |
| <code>\vcenter</code> .....                  | 458                |
| <code>\verbatim-file</code> .....            | 473                |
| <code>\virga</code> .....                    | 276                |
| <code>\virgula</code> .....                  | 269                |
| <code>\voiceFourStyle</code> .....           | 105                |
| <code>\voiceNeutralStyle</code> .....        | 105                |
| <code>\voiceOne</code> .....                 | 103                |
| <code>\voiceOne ... \voiceFour</code> .....  | 103                |
| <code>\voiceOneStyle</code> .....            | 105                |
| <code>\voiceThreeStyle</code> .....          | 105                |
| <code>\voiceTwoStyle</code> .....            | 105                |
| <code>\whiteout</code> .....                 | 474                |
| <code>\with</code> .....                     | 358                |
| <code>\with-color</code> .....               | 146, 474           |
| <code>\with-dimensions</code> .....          | 474                |
| <code>\with-url</code> .....                 | 464                |
| <code>\wordwrap</code> .....                 | 166, 459           |
| <code>\wordwrap-field</code> .....           | 458                |
| <code>\wordwrap-internal</code> .....        | 475                |
| <code>\wordwrap-lines</code> .....           | 475                |
| <code>\wordwrap-string</code> .....          | 459                |
| <code>\wordwrap-string-internal</code> ..... | 475                |

|  
| ..... 67, 68

~  
~ ..... 35

## A

|                                           |     |
|-------------------------------------------|-----|
| <code>after-title-space</code> .....      | 316 |
| <code>annotate-spacing</code> .....       | 351 |
| <code>arpeggio</code> .....               | 88  |
| <code>arpeggioArrowDown</code> .....      | 88  |
| <code>arpeggioArrowUp</code> .....        | 88  |
| <code>arpeggioBracket</code> .....        | 89  |
| <code>arpeggioNormal</code> .....         | 88  |
| <code>arpeggioParenthesis</code> .....    | 89  |
| <code>arranger</code> .....               | 290 |
| <code>aug</code> .....                    | 242 |
| <code>auto-first-page-number</code> ..... | 317 |
| <code>autoBeaming</code> .....            | 60  |
| <code>autoBeamSettings</code> .....       | 57  |
| <code>autochange</code> .....             | 192 |

## B

|                                           |         |
|-------------------------------------------|---------|
| <code>Balloon_engraver</code> .....       | 149     |
| <code>balloonGrobText</code> .....        | 149     |
| <code>balloonLengthOff</code> .....       | 149     |
| <code>balloonLengthOn</code> .....        | 149     |
| <code>balloonText</code> .....            | 149     |
| <code>barCheckSynchronize</code> .....    | 67      |
| <code>barNumberVisibility</code> .....    | 64      |
| <code>base-shortest-duration</code> ..... | 341     |
| <code>before-title-space</code> .....     | 316     |
| <code>bendAfter</code> .....              | 86      |
| <code>between-system-padding</code> ..... | 315     |
| <code>between-system-space</code> .....   | 315     |
| <code>between-title-space</code> .....    | 316     |
| <code>blank-last-page-force</code> .....  | 317     |
| <code>blank-page-force</code> .....       | 317     |
| <code>bookTitleMarkup</code> .....        | 293     |
| <code>bottom-margin</code> .....          | 315     |
| <code>bracket</code> .....                | 82, 196 |
| <code>breakable</code> .....              | 56      |
| <code>breakbefore</code> .....            | 290     |
| <code>breathe</code> .....                | 85      |

## C

|                                             |        |
|---------------------------------------------|--------|
| <code>change</code> .....                   | 191    |
| <code>chordNameExceptions</code> .....      | 249    |
| <code>chordNameSeparator</code> .....       | 249    |
| <code>chordNoteNamer</code> .....           | 249    |
| <code>chordPrefixSpacer</code> .....        | 249    |
| <code>chordRootNamer</code> .....           | 249    |
| <code>color</code> .....                    | 146    |
| <code>common-shortest-duration</code> ..... | 341    |
| <code>composer</code> .....                 | 290    |
| <code>controlpitch</code> .....             | 7      |
| <code>copyright</code> .....                | 290    |
| <code>crescHairpin</code> .....             | 78, 79 |
| <code>crescTextCresc</code> .....           | 78, 79 |
| <code>cross</code> .....                    | 26     |
| <code>cross-staff</code> .....              | 194    |
| <code>currentBarNumber</code> .....         | 64, 75 |

## D

|                                   |        |
|-----------------------------------|--------|
| <code>dedication</code> .....     | 290    |
| <code>default</code> .....        | 18, 20 |
| <code>defaultBarType</code> ..... | 64     |

dim ..... 242  
 dimHairpin ..... 78, 79  
 dimTextDecr ..... 78, 79  
 dimTextDecresc ..... 78, 79  
 dimTextDim ..... 78, 79  
 dodecaphonic ..... 22  
 dynamic ..... 82  
 dynamicDown ..... 79  
 DynamicLineSpanner ..... 79  
 dynamicNeutral ..... 79  
 dynamicUp ..... 79

## E

espressivo ..... 78  
 evenFooterMarkup ..... 293  
 evenHeaderMarkup ..... 293

## F

f ..... 78  
 ff ..... 78  
 fff ..... 78  
 ffff ..... 78  
 finger ..... 143  
 first-page-number ..... 314  
 flag-style ..... 194  
 followVoice ..... 194  
 font-interface ..... 143  
 font-interface ..... 171  
 font-size ..... 142, 143  
 fontSize ..... 142  
 foot-separation ..... 315  
 forget ..... 23  
 fp ..... 78

## G

glissando ..... 87  
 Grid\_line\_span\_engraver ..... 150  
 Grid\_point\_engraver ..... 150  
 gridInterval ..... 150

## H

head-separation ..... 315  
 hideKeySignature ..... 238  
 hideNotes ..... 145  
 hideStaffSwitch ..... 194  
 horizontal-shift ..... 316  
 Horizontal\_bracket\_engraver ..... 152  
 huge ..... 142

## I

indent ..... 344  
 instrument ..... 290

## L

large ..... 142  
 layout file ..... 319  
 left-margin ..... 315  
 length ..... 194

line-width ..... 315, 344  
 ly:minimal-breaking ..... 323  
 ly:optimal-breaking ..... 322  
 ly:page-turn-breaking ..... 322

## M

m ..... 242  
 magstep ..... 142  
 maj ..... 242  
 major seven symbols ..... 249  
 majorSevenSymbol ..... 249  
 make-dynamic-script ..... 82  
 makeClusters ..... 102  
 measureLength ..... 75  
 measurePosition ..... 75  
 mergeDifferentlyDottedOff ..... 106  
 mergeDifferentlyDottedOn ..... 106  
 mergeDifferentlyHeadedOff ..... 106  
 mergeDifferentlyHeadedOn ..... 106  
 meter ..... 290  
 mf ..... 78  
 minimumFret ..... 204  
 minimumPageTurnLength ..... 323  
 minimumRepeatLengthForPageTurn ..... 323  
 mixed ..... 196  
 modern ..... 20  
 modern-cautionary ..... 21  
 modern-voice ..... 21  
 modern-voice-cautionary ..... 21  
 mp ..... 78

## N

neo-modern ..... 22  
 neo-modern-cautionary ..... 22  
 no-reset ..... 23  
 normalsize ..... 142

## O

oddFooterMarkup ..... 293  
 oddHeaderMarkup ..... 293  
 oneVoice ..... 103  
 opus ..... 290  
 outside-staff-horizontal-padding ..... 339  
 outside-staff-padding ..... 339  
 outside-staff-priority ..... 339

## P

p ..... 78  
 page-breaking-between-system-padding ..... 316  
 page-spacing-weight ..... 317  
 page-top-space ..... 315  
 paper-height ..... 315  
 paper-width ..... 315  
 papersize ..... 314  
 parallelMusic ..... 113  
 parenthesesize ..... 147  
 partcombine ..... 109  
 pedalSustainStyle ..... 196  
 percent ..... 99  
 phrasingSlurDown ..... 85

|                         |          |
|-------------------------|----------|
| phrasingSlurNeutral     | 85       |
| phrasingSlurUp          | 85       |
| piano                   | 21       |
| piano-cautionary        | 22       |
| PianoStaff              | 190, 192 |
| piece                   | 290      |
| pipeSymbol              | 68       |
| pitchedTrill            | 91       |
| poet                    | 290      |
| pp                      | 78       |
| ppp                     | 78       |
| pppp                    | 78       |
| ppppp                   | 78       |
| print-first-page-number | 314      |
| print-page-number       | 314      |
| printallheaders         | 292, 316 |

## R

|                    |     |
|--------------------|-----|
| r                  | 38  |
| R                  | 41  |
| ragged-bottom      | 315 |
| ragged-last        | 344 |
| ragged-last-bottom | 315 |
| ragged-right       | 344 |
| relative           | 193 |
| repeatCommands     | 96  |
| rfz                | 78  |
| rgb-color          | 146 |

## S

|                      |     |
|----------------------|-----|
| s                    | 40  |
| scoreTitleMarkup     | 293 |
| set-accidental-style | 18  |
| set-octavation       | 16  |
| sf                   | 78  |
| sff                  | 78  |
| sfz                  | 78  |
| shiftOff             | 106 |
| shiftOn              | 106 |
| shiftOnn             | 106 |
| shiftOnnn            | 106 |
| showKeySignature     | 238 |
| showLastLength       | 303 |
| showStaffSwitch      | 194 |
| skipTypesetting      | 303 |
| slurDashed           | 84  |
| slurDotted           | 84  |
| slurDown             | 83  |
| slurNeutral          | 83  |
| slurSolid            | 84  |
| small                | 142 |
| sostenutoOff         | 196 |
| sostenutoOn          | 196 |
| sp                   | 78  |
| spacing              | 341 |
| spp                  | 78  |
| staff-padding        | 191 |
| Staff.midiInstrument | 305 |
| start-repeat         | 96  |

|                         |     |
|-------------------------|-----|
| startGroup              | 152 |
| startTrillSpan          | 91  |
| Stem                    | 194 |
| stem-spacing-correction | 341 |
| stemLeftBeamCount       | 60  |
| stemRightBeamCount      | 60  |
| stopGroup               | 152 |
| stopTrillSpan           | 91  |
| subdivideBeams          | 56  |
| subsubtitle             | 290 |
| subtitle                | 290 |
| suggestAccidentals      | 279 |
| sus                     | 245 |
| sustainOff              | 196 |
| sustainOn               | 196 |
| system-count            | 315 |
| systemSeparatorMarkup   | 316 |

## T

|                            |     |
|----------------------------|-----|
| tagline                    | 290 |
| taor                       | 238 |
| teeny                      | 142 |
| text                       | 196 |
| textSpannerDown            | 155 |
| textSpannerNeutral         | 155 |
| textSpannerUp              | 155 |
| thumb                      | 144 |
| tiny                       | 142 |
| title                      | 290 |
| top-margin                 | 315 |
| treCorde                   | 196 |
| tremolo                    | 100 |
| tremoloFlags               | 101 |
| trill                      | 91  |
| TupletNumber               | 33  |
| tupletNumberFormatFunction | 32  |
| tupletSpannerDuration      | 32  |

## U

|             |     |
|-------------|-----|
| unaCorda    | 196 |
| unfold      | 98  |
| unHideNotes | 145 |

## V

|          |        |
|----------|--------|
| voice    | 18, 20 |
| Voice    | 103    |
| voiceOne | 103    |

## W

|            |     |
|------------|-----|
| whichBar   | 64  |
| with-color | 146 |

## X

|           |     |
|-----------|-----|
| x11-color | 146 |
| x11-color | 147 |

## Anhang F LilyPond-Index

Zusätzlich zu allen LilyPond Befehlen und Schlüsselwörtern listet dieser Index alle relevanten Begriffe auf und verlinkt sie mit den entsprechenden Abschnitten, wo sie erklärt werden. Der erste Teil zeigt auf die genaue Stelle im Handbuch, an der der Begriff vorkommt, der zweite Teil zeigt auf den gesamten Abschnitt, in dem das Thema behandelt wird.

|                                  |     |                                                      |     |
|----------------------------------|-----|------------------------------------------------------|-----|
| !                                |     | \bold.....                                           | 135 |
| !.....                           | 4   | \box.....                                            | 135 |
| ,                                |     | \bracket.....                                        | 157 |
| '.....                           | 2   | \breve.....                                          | 11  |
| ,                                |     | \cadenzaOff.....                                     | 20  |
| ,                                |     | \cadenzaOn.....                                      | 20  |
| ,                                |     | \caesura.....                                        | 94  |
| ,                                |     | \caps.....                                           | 136 |
| ,                                |     | \cavum.....                                          | 101 |
| ,                                | 2   | \center-align.....                                   | 144 |
| .                                |     | \center-column.....                                  | 144 |
| .                                |     | \char.....                                           | 168 |
| .                                | 11  | \circle.....                                         | 158 |
| .                                |     | \clef.....                                           | 8   |
| /                                |     | \column.....                                         | 144 |
| /.....                           | 79  | \column-lines.....                                   | 171 |
| /+.....                          | 79  | \combine.....                                        | 145 |
| ?                                |     | \concat.....                                         | 145 |
| ?.....                           | 4   | \deminutum.....                                      | 101 |
| [                                |     | \descendens.....                                     | 101 |
| [.....                           | 22  | \dir-column.....                                     | 145 |
| ]                                |     | \divisioMaior.....                                   | 94  |
| ]                                |     | \divisioMaxima.....                                  | 94  |
| ]                                |     | \divisioMinima.....                                  | 94  |
| ]                                |     | \dorian.....                                         | 9   |
| ]                                |     | \dotsDown.....                                       | 12  |
| ]                                |     | \dotsNeutral (Wiederherstellung des Standards.)..... | 12  |
| ]                                |     | \dotsUp.....                                         | 12  |
| ]                                |     | \doubleflat.....                                     | 161 |
| ]                                |     | \doublesharp.....                                    | 161 |
| ]                                | 22  | \draw-circle.....                                    | 158 |
| ]                                |     | \draw-line.....                                      | 158 |
| \                                |     | \dynamic.....                                        | 136 |
| \!                               | 32  | \dynamicDown.....                                    | 34  |
| \<                               | 32  | \dynamicNeutral.....                                 | 34  |
| \>                               | 32  | \dynamicUp.....                                      | 34  |
| \\                               | 45  | \epsfile.....                                        | 158 |
| \abs-fontsize.....               | 135 | \f.....                                              | 32  |
| \aeolian.....                    | 9   | \ff.....                                             | 32  |
| \afterGrace.....                 | 27  | \fff.....                                            | 32  |
| \alternative notieren, etwa..... | 40  | \ffff.....                                           | 32  |
| \arpeggio.....                   | 37  | \fill-line.....                                      | 146 |
| \arpeggioArrowDown.....          | 38  | \filled-box.....                                     | 159 |
| \arpeggioArrowUp.....            | 38  | \finalis.....                                        | 94  |
| \arpeggioBracket.....            | 38  | \finger.....                                         | 136 |
| \arpeggioNormal.....             | 38  | \flat.....                                           | 162 |
| \arrow-head.....                 | 157 | \flexa.....                                          | 101 |
| \ascendens.....                  | 101 | \fontCaps.....                                       | 136 |
| \auctum.....                     | 101 | \fontsize.....                                       | 136 |
| \augmentum.....                  | 101 | \fp.....                                             | 32  |
| \backslashed-digit.....          | 167 | \fraction.....                                       | 168 |
| \bar.....                        | 24  | \frenchChords.....                                   | 83  |
| \beam.....                       | 157 | \fret-diagram.....                                   | 165 |
| \bigger.....                     | 135 | \fret-diagram-terse.....                             | 165 |
|                                  |     | \fret-diagram-verbose.....                           | 166 |

|                                       |     |                                         |        |
|---------------------------------------|-----|-----------------------------------------|--------|
| <code>\fromproperty</code> .....      | 168 | <code>\phrasingSlurDown</code> .....    | 35     |
| <code>\general-align</code> .....     | 146 | <code>\phrasingSlurNeutral</code> ..... | 35     |
| <code>\germanChords</code> .....      | 83  | <code>\phrasingSlurUp</code> .....      | 35     |
| <code>\glissando</code> .....         | 36  | <code>\phrygian</code> .....            | 9      |
| <code>\grace</code> .....             | 26  | <code>\postscript</code> .....          | 159    |
| <code>\halign</code> .....            | 147 | <code>\pp</code> .....                  | 32     |
| <code>\harp-pedal</code> .....        | 167 | <code>\ppp</code> .....                 | 32     |
| <code>\hbracket</code> .....          | 159 | <code>\pppp</code> .....                | 32     |
| <code>\hcenter-in</code> .....        | 148 | <code>\put-adjacent</code> .....        | 153    |
| <code>\hideStaffSwitch</code> .....   | 63  | <code>\quilisma</code> .....            | 101    |
| <code>\hspace</code> .....            | 148 | <code>\raise</code> .....               | 153    |
| <code>\huge</code> .....              | 137 | <code>\relative</code> .....            | 2      |
| <code>\inclinatum</code> .....        | 101 | <code>\repeat</code> .....              | 40     |
| <code>\ionian</code> .....            | 9   | <code>\repeatTie</code> .....           | 15, 41 |
| <code>\italianChords</code> .....     | 83  | <code>\rest</code> .....                | 17     |
| <code>\italic</code> .....            | 137 | <code>\rfz</code> .....                 | 32     |
| <code>\justified-lines</code> .....   | 172 | <code>\right-align</code> .....         | 153    |
| <code>\justify</code> .....           | 149 | <code>\right-column</code> .....        | 153    |
| <code>\justify-field</code> .....     | 149 | <code>\roman</code> .....               | 140    |
| <code>\justify-string</code> .....    | 150 | <code>\rotate</code> .....              | 154    |
| <code>\key</code> .....               | 9   | <code>\rounded-box</code> .....         | 160    |
| <code>\large</code> .....             | 137 | <code>\sans</code> .....                | 140    |
| <code>\larger</code> .....            | 137 | <code>\score</code> .....               | 163    |
| <code>\left-align</code> .....        | 150 | <code>\semiflat</code> .....            | 164    |
| <code>\left-column</code> .....       | 151 | <code>\semiGermanChords</code> .....    | 83     |
| <code>\line</code> .....              | 151 | <code>\semisharp</code> .....           | 164    |
| <code>\linea</code> .....             | 101 | <code>\sesquiflat</code> .....          | 164    |
| <code>\locrian</code> .....           | 9   | <code>\sesquisharp</code> .....         | 164    |
| <code>\longa</code> .....             | 11  | <code>\sf</code> .....                  | 32     |
| <code>\lookup</code> .....            | 168 | <code>\sff</code> .....                 | 32     |
| <code>\lower</code> .....             | 151 | <code>\sfz</code> .....                 | 32     |
| <code>\lydian</code> .....            | 9   | <code>\sharp</code> .....               | 164    |
| <code>\magnify</code> .....           | 138 | <code>\shiftOff</code> .....            | 50     |
| <code>\major</code> .....             | 9   | <code>\shiftOn</code> .....             | 50     |
| <code>\markalphabet</code> .....      | 169 | <code>\shiftOnn</code> .....            | 50     |
| <code>\markletter</code> .....        | 169 | <code>\shiftOnnn</code> .....           | 50     |
| <code>\maxima</code> .....            | 11  | <code>\showStaffSwitch</code> .....     | 63     |
| <code>\medium</code> .....            | 138 | <code>\simple</code> .....              | 140    |
| <code>\mf</code> .....                | 32  | <code>\skip</code> .....                | 17     |
| <code>\minor</code> .....             | 9   | <code>\slashed-digit</code> .....       | 170    |
| <code>\mixolydian</code> .....        | 9   | <code>\slurDashed</code> .....          | 35     |
| <code>\mp</code> .....                | 32  | <code>\slurDotted</code> .....          | 35     |
| <code>\musicglyph</code> .....        | 162 | <code>\slurDown</code> .....            | 35     |
| <code>\natural</code> .....           | 162 | <code>\slurNeutral</code> .....         | 35     |
| <code>\normal-size-sub</code> .....   | 138 | <code>\slurSolid</code> .....           | 35     |
| <code>\normal-size-super</code> ..... | 139 | <code>\slurUp</code> .....              | 35     |
| <code>\normal-text</code> .....       | 139 | <code>\small</code> .....               | 141    |
| <code>\normalsize</code> .....        | 139 | <code>\smallCaps</code> .....           | 141    |
| <code>\note</code> .....              | 162 | <code>\smaller</code> .....             | 141    |
| <code>\note-by-number</code> .....    | 162 | <code>\sp</code> .....                  | 32     |
| <code>\null</code> .....              | 169 | <code>\spp</code> .....                 | 32     |
| <code>\number</code> .....            | 140 | <code>\startTrillSpan</code> .....      | 39     |
| <code>\on-the-fly</code> .....        | 169 | <code>\stemDown</code> .....            | 58     |
| <code>\oneVoice</code> .....          | 50  | <code>\stemNeutral</code> .....         | 58     |
| <code>\oriscus</code> .....           | 101 | <code>\stemUp</code> .....              | 58     |
| <code>\override</code> .....          | 169 | <code>\stencil</code> .....             | 170    |
| <code>\override-lines</code> .....    | 172 | <code>\stopTrillSpan</code> .....       | 39     |
| <code>\p</code> .....                 | 32  | <code>\strophia</code> .....            | 101    |
| <code>\pad-around</code> .....        | 151 | <code>\strut</code> .....               | 170    |
| <code>\pad-markup</code> .....        | 152 | <code>\sub</code> .....                 | 141    |
| <code>\pad-to-box</code> .....        | 152 | <code>\super</code> .....               | 142    |
| <code>\pad-x</code> .....             | 152 | <code>\teeny</code> .....               | 142    |
| <code>\page-ref</code> .....          | 170 | <code>\text</code> .....                | 142    |
| <code>\partial</code> .....           | 19  | <code>\tied-lyric</code> .....          | 165    |
| <code>\pes</code> .....               | 101 | <code>\tieDashed</code> .....           | 16     |

|                                              |     |
|----------------------------------------------|-----|
| <code>\tieDotted</code> .....                | 16  |
| <code>\tieDown</code> .....                  | 16  |
| <code>\tieNeutral</code> .....               | 16  |
| <code>\tieSolid</code> .....                 | 16  |
| <code>\tieUp</code> .....                    | 16  |
| <code>\time</code> .....                     | 18  |
| <code>\times</code> .....                    | 12  |
| <code>\tiny</code> .....                     | 142 |
| <code>\translate</code> .....                | 154 |
| <code>\translate-scaled</code> .....         | 154 |
| <code>\transparent</code> .....              | 170 |
| <code>\transpose</code> .....                | 7   |
| <code>\triangle</code> .....                 | 161 |
| <code>\tupletDown</code> .....               | 12  |
| <code>\tupletNeutral</code> .....            | 12  |
| <code>\tupletUp</code> .....                 | 12  |
| <code>\typewriter</code> .....               | 143 |
| <code>\underline</code> .....                | 143 |
| <code>\unfoldRepeats</code> .....            | 105 |
| <code>\upright</code> .....                  | 143 |
| <code>\vcenter</code> .....                  | 155 |
| <code>\verbatim-file</code> .....            | 170 |
| <code>\virga</code> .....                    | 101 |
| <code>\virgula</code> .....                  | 94  |
| <code>\voiceFour</code> .....                | 50  |
| <code>\voiceOne</code> .....                 | 50  |
| <code>\voiceThree</code> .....               | 50  |
| <code>\voiceTwo</code> .....                 | 50  |
| <code>\whiteout</code> .....                 | 171 |
| <code>\with-color</code> .....               | 171 |
| <code>\with-dimensions</code> .....          | 171 |
| <code>\with-url</code> .....                 | 161 |
| <code>\wordwrap</code> .....                 | 156 |
| <code>\wordwrap-field</code> .....           | 155 |
| <code>\wordwrap-internal</code> .....        | 172 |
| <code>\wordwrap-lines</code> .....           | 172 |
| <code>\wordwrap-string</code> .....          | 156 |
| <code>\wordwrap-string-internal</code> ..... | 172 |

|

|       |    |
|-------|----|
| ..... | 25 |
|-------|----|

~

|         |    |
|---------|----|
| ~ ..... | 15 |
|---------|----|

## A

|                                                 |          |
|-------------------------------------------------|----------|
| Abstand zwischen Systemen in Klaviernoten ..... | 63       |
| Abstrich .....                                  | 30, 172  |
| acciaccatura .....                              | 112, 196 |
| Accidental .....                                | 87       |
| Accidental_engraver .....                       | 104      |
| Accidentals and key signatures .....            | 4, 9     |
| AccidentalSuggestion .....                      | 104      |
| addChordShape .....                             | 112, 196 |
| adding a white background to text .....         | 171      |
| addInstrumentDefinition .....                   | 112, 196 |
| Additionen in Akkorden .....                    | 78       |
| addQuote .....                                  | 112, 196 |
| afterGrace .....                                | 112, 196 |
| Akkord-Modus .....                              | 77       |
| Akkord-Notation .....                           | 77       |

|                                         |          |
|-----------------------------------------|----------|
| Akkordbezeichnungen .....               | 77       |
| Akkorddiagramm .....                    | 68       |
| Akkorde .....                           | 45, 80   |
| Akkorde, Jazz .....                     | 83       |
| Akkordmodi .....                        | 79       |
| Akkordsymbole .....                     | 80       |
| Akkordtabulatur .....                   | 68       |
| Akzent .....                            | 30, 172  |
| Akzidentien .....                       | 3        |
| Al niente .....                         | 32       |
| allowPageTurn .....                     | 112, 196 |
| Altschlüssel .....                      | 8        |
| Anzahl der Notenlinien einstellen ..... | 55       |
| applyContext .....                      | 112, 196 |
| applyMusic .....                        | 112, 196 |
| applyOutput .....                       | 112, 196 |
| appoggiatura .....                      | 112, 196 |
| Arpeggio .....                          | 37       |
| Arpeggio .....                          | 38       |
| Arten von Notenköpfen .....             | 134      |
| articulations .....                     | 92       |
| Artikulationszeichen .....              | 29       |
| assertBeamQuant .....                   | 112, 196 |
| assertBeamSlope .....                   | 112, 197 |
| Aufstrich .....                         | 30, 172  |
| Auftakt .....                           | 19       |
| aug .....                               | 79       |
| Ausgabe von Akkordbezeichnungen .....   | 80       |
| Ausklängen lassen .....                 | 35       |
| Ausnahmen, Akkordsymbole .....          | 82       |
| auto-knee-gap .....                     | 23       |
| autochange .....                        | 112, 197 |
| AutoChangeMusic .....                   | 62       |
| Automatischer Systemwechsel .....       | 61       |

## B

|                                 |          |
|---------------------------------|----------|
| backslashed digits .....        | 167      |
| Balken und Zeilenumbrüche ..... | 23       |
| Balken zwischen Systemen .....  | 23       |
| Balken, manuell .....           | 22       |
| balloonGrobText .....           | 112, 197 |
| balloonText .....               | 113, 197 |
| Banjo tablatures .....          | 70       |
| Banter .....                    | 83       |
| bar .....                       | 113, 197 |
| bar-line-interface .....        | 182      |
| Bar_engraver .....              | 80       |
| barCheckSynchronize .....       | 25       |
| Baritonschlüssel .....          | 8        |
| BarLine .....                   | 25       |
| barNumberCheck .....            | 113, 197 |
| BassFigure .....                | 84, 86   |
| BassFigureAlignment .....       | 86       |
| BassFigureBracket .....         | 86       |
| BassFigureContinuation .....    | 86       |
| BassFigureLine .....            | 86       |
| Basso continuo .....            | 83       |
| Bassschlüssel .....             | 8        |
| Beam .....                      | 22, 44   |
| Beginn eines Notensystems ..... | 53       |
| bendAfter .....                 | 113, 197 |
| Beschriftung .....              | 29       |
| Bezifferter Bass .....          | 83       |

|                             |          |
|-----------------------------|----------|
| Bindebogen .....            | 15       |
| Bögen, laissez vibrer ..... | 35       |
| Bratschenschlüssel .....    | 8        |
| <b>breakable</b> .....      | 23       |
| <b>breathe</b> .....        | 113, 197 |
| <b>BreathingSign</b> .....  | 36, 94   |
| Bund .....                  | 66       |

## C

|                                          |          |
|------------------------------------------|----------|
| C-Schlüssel .....                        | 8        |
| centering a column of text .....         | 144      |
| changing direction of text columns ..... | 145      |
| <b>ChoirStaff</b> .....                  | 54       |
| <b>chordNameExceptions</b> .....         | 81       |
| <b>ChordNames</b> .....                  | 80       |
| <b>chordNameSeparator</b> .....          | 82       |
| <b>chordNoteNamer</b> .....              | 82       |
| <b>chordPrefixSpacer</b> .....           | 82       |
| <b>chordRootNamer</b> .....              | 82       |
| circling text .....                      | 158      |
| <b>clef</b> .....                        | 113, 197 |
| <b>Clef</b> .....                        | 9        |
| Cluster .....                            | 79       |
| Coda .....                               | 30, 172  |
| coloring text .....                      | 171      |
| <b>Completion_heads_engraver</b> .....   | 21       |
| concatenating text .....                 | 145      |
| controlling general text alignment ..... | 146      |
| creating empty text objects .....        | 169      |
| creating horizontal spaces in text ..... | 148      |
| creating text fractions .....            | 168      |
| creating vertical spaces in text .....   | 170      |
| Crescendo .....                          | 33       |
| <b>cueDuring</b> .....                   | 113, 197 |
| Custodes .....                           | 93       |
| Custos .....                             | 93       |
| <b>Custos_engraver</b> .....             | 93       |

## D

|                                                      |          |
|------------------------------------------------------|----------|
| Dauer .....                                          | 11       |
| Daumenbezeichnung .....                              | 30, 172  |
| Decrescendo .....                                    | 33       |
| <b>defaultBarType</b> .....                          | 25       |
| Dicke der Notenlinien einstellen .....               | 55       |
| Die Feta-Schriftart .....                            | 162      |
| <b>dim</b> .....                                     | 79       |
| Diminuendo .....                                     | 33       |
| <b>dispatcher</b> .....                              | 199      |
| <b>displayLilyMusic</b> .....                        | 113, 197 |
| <b>displayMusic</b> .....                            | 113, 197 |
| divisio .....                                        | 94       |
| divisiones .....                                     | 94       |
| Doppellinie .....                                    | 24       |
| Doppelpraller .....                                  | 30, 172  |
| <b>DotColumn</b> .....                               | 12       |
| <b>Dots</b> .....                                    | 12       |
| <b>DoublePercentRepeat</b> .....                     | 43       |
| <b>DoublePercentRepeatCounter</b> .....              | 43       |
| drawing beams within text .....                      | 157      |
| drawing boxes with rounded corners .....             | 159      |
| drawing boxes with rounded corners around text ..... | 160      |

|                                       |            |
|---------------------------------------|------------|
| drawing circles within text .....     | 158        |
| drawing lines within text .....       | 158        |
| drawing solid boxes within text ..... | 159        |
| drawing triangles within text .....   | 161        |
| Drums .....                           | 71         |
| <b>DrumStaff</b> .....                | 71, 74     |
| <b>DrumVoice</b> .....                | 71, 72, 74 |
| <b>DynamicLineSpanner</b> .....       | 33         |
| <b>DynamicLineSpanner</b> .....       | 34         |
| <b>DynamicText</b> .....              | 34         |
| Dynamik .....                         | 32         |

## E

|                                                   |          |
|---------------------------------------------------|----------|
| Eingabe von Akkorden .....                        | 77       |
| Eingabe von Noten parallel .....                  | 51       |
| Einstellung von automatischen Balken .....        | 174      |
| enclosing text in a box with rounded corners .... | 160      |
| enclosing text within a box .....                 | 135      |
| <b>endSpanners</b> .....                          | 113, 197 |
| Entfernen von Tönen aus Akkorden .....            | 78       |
| Erinnerungsvorzeichen .....                       | 4        |
| Erweiterung von Wiederholungen .....              | 105      |
| Espressivo .....                                  | 30, 172  |

## F

|                                           |             |
|-------------------------------------------|-------------|
| Fähnchen, Alte Musik .....                | 90          |
| Farben, Liste .....                       | 119         |
| FDL, GNU Free Documentation License ..... | 222         |
| <b>featherDurations</b> .....             | 113, 197    |
| Fermate .....                             | 30, 172     |
| Feta font .....                           | 121         |
| <b>FiguredBass</b> .....                  | 84          |
| <b>FiguredBass</b> .....                  | 85          |
| <b>FiguredBass</b> .....                  | 86          |
| finalis .....                             | 94          |
| <b>Fingering</b> .....                    | 57          |
| Fingersatz .....                          | 30, 56, 172 |
| Fingersatz, rechte Hand, Gitarre .....    | 69          |
| Fingerwechsel .....                       | 56          |
| Flageolet .....                           | 30, 172     |
| Flageolett .....                          | 65          |
| <b>followVoice</b> .....                  | 62          |
| Font, Feta .....                          | 121         |
| <b>font-interface</b> .....               | 169         |
| <b>Forbid_line_break_engraver</b> .....   | 21          |
| Formatierung von Brüchen .....            | 13          |
| Formatierung von Triolen .....            | 13          |
| Franz. Violinschlüssel .....              | 8           |
| <b>fret-diagram-interface</b> .....       | 69          |
| Fret-Diagramme .....                      | 68          |
| Fußbezeichnung .....                      | 30, 172     |

## G

|                            |          |
|----------------------------|----------|
| Gebrochene Akkorde .....   | 37       |
| Gedämpft .....             | 30, 172  |
| Generalbass .....          | 83       |
| Gitarrengriffsymbole ..... | 68       |
| Glissando .....            | 36       |
| <b>Glissando</b> .....     | 37       |
| <b>grace</b> .....         | 113, 197 |
| <b>GraceMusic</b> .....    | 28       |



|                                                  |    |
|--------------------------------------------------|----|
| GrandStaff .....                                 | 53 |
| Gregorianische quadratische Neumenligaturen .... | 96 |
| Griffsymbole .....                               | 68 |
| Grundton eines Akkordes.....                     | 78 |

## H

|                                        |     |
|----------------------------------------|-----|
| Hairpin.....                           | 32  |
| Hairpin.....                           | 34  |
| Hälse über zwei Systeme .....          | 63  |
| Hinzufügen von Tönen in Akkorden ..... | 78  |
| horizontally centering text .....      | 144 |

## I

|                                                |          |
|------------------------------------------------|----------|
| importing stencils into text.....              | 170      |
| includePageLayoutFile.....                     | 113, 197 |
| inlining an Encapsulated PostScript image..... | 158      |
| inserting music into text .....                | 163      |
| inserting PostScript directly into text .....  | 159      |
| inserting URL links into text .....            | 161      |
| Instrumentengruppe.....                        | 53       |
| instrumentSwitch.....                          | 113, 197 |

## J

|                                    |     |
|------------------------------------|-----|
| Jazz-Akkorde .....                 | 83  |
| Justierung von Notensystemen ..... | 55  |
| justifying lines of text.....      | 172 |
| justifying text.....               | 149 |

## K

|                              |          |
|------------------------------|----------|
| Kadenz .....                 | 20       |
| keepWithTag.....             | 113, 197 |
| KeyCancellation .....        | 10       |
| KeySignature .....           | 10, 87   |
| killCues .....               | 113, 197 |
| Kirchentonarten.....         | 9        |
| Klammer, geschweift .....    | 53       |
| Klammer, vertikal.....       | 53       |
| Klammer, Wiederholung .....  | 40       |
| Klammern um Vorzeichen ..... | 4        |

## L

|                                                 |          |
|-------------------------------------------------|----------|
| label.....                                      | 113, 197 |
| Laissez vibrer .....                            | 35       |
| LaissezVibrerTie .....                          | 35       |
| LaissezVibrerTieColumn .....                    | 35       |
| Lautstärke .....                                | 32       |
| Layers .....                                    | 46       |
| LedgerLineSpanner .....                         | 5        |
| left aligning text.....                         | 150      |
| Legatobögen .....                               | 34       |
| Ligature_bracket_engraver.....                  | 95, 96   |
| LigatureBracket .....                           | 94       |
| Ligaturen .....                                 | 94       |
| Ligaturen der quadratischen Neumennotation..... | 96       |
| Ligaturen, weiße Mensuralnotation .....         | 95       |
| ligatures in text.....                          | 145      |
| Liste der Farben .....                          | 119      |
| listener .....                                  | 199      |
| lowering text.....                              | 151      |

|                                        |     |
|----------------------------------------|-----|
| ly:add-file-name-alist .....           | 199 |
| ly:add-interface .....                 | 199 |
| ly:add-listener .....                  | 199 |
| ly:add-option.....                     | 199 |
| ly:all-grob-interfaces .....           | 199 |
| ly:all-options .....                   | 199 |
| ly:all-stencil-expressions.....        | 199 |
| ly:assoc-get.....                      | 199 |
| ly:book-add-score!.....                | 199 |
| ly:book-process .....                  | 200 |
| ly:book-process-to-systems.....        | 200 |
| ly:box? .....                          | 200 |
| ly:bp.....                             | 200 |
| ly:bracket .....                       | 200 |
| ly:broadcast.....                      | 200 |
| ly:camel-case->lisp-identifier .....   | 200 |
| ly:chain-assoc-get.....                | 200 |
| ly:clear-anonymous-modules.....        | 200 |
| ly:cm .....                            | 200 |
| ly:command-line-code .....             | 200 |
| ly:command-line-options .....          | 200 |
| ly:command-line-verbose?.....          | 200 |
| ly:connect-dispatchers .....           | 200 |
| ly:context-event-source .....          | 200 |
| ly:context-events-below.....           | 200 |
| ly:context-find .....                  | 200 |
| ly:context-grob-definition.....        | 201 |
| ly:context-id.....                     | 201 |
| ly:context-name .....                  | 201 |
| ly:context-now .....                   | 201 |
| ly:context-parent.....                 | 201 |
| ly:context-property .....              | 201 |
| ly:context-property-where-defined..... | 201 |
| ly:context-pushpop-property .....      | 201 |
| ly:context-set-property!.....          | 201 |
| ly:context-unset-property.....         | 201 |
| ly:context?.....                       | 201 |
| ly:default-scale .....                 | 201 |
| ly:dimension?.....                     | 201 |
| ly:dir? .....                          | 201 |
| ly:duration->string.....               | 201 |
| ly:duration-dot-count .....            | 201 |
| ly:duration-factor .....               | 201 |
| ly:duration-length.....                | 201 |
| ly:duration-log .....                  | 201 |
| ly:duration<?.....                     | 202 |
| ly:duration?.....                      | 202 |
| ly:effective-prefix.....               | 202 |
| ly:error .....                         | 202 |
| ly:eval-simple-closure .....           | 202 |
| ly:event-deep-copy.....                | 202 |
| ly:event-property .....                | 202 |
| ly:event-set-property! .....           | 202 |
| ly:expand-environment .....            | 202 |
| ly:export .....                        | 202 |
| ly:find-accidentals-simple.....        | 202 |
| ly:find-file.....                      | 202 |
| ly:font-config-display-fonts .....     | 202 |
| ly:font-config-get-font-file .....     | 202 |
| ly:font-design-size.....               | 202 |
| ly:font-file-name .....                | 202 |
| ly:font-get-glyph.....                 | 202 |
| ly:font-glyph-name-to-charcode .....   | 203 |
| ly:font-glyph-name-to-index .....      | 203 |
| ly:font-index-to-charcode.....         | 203 |



|                                        |     |                                       |     |
|----------------------------------------|-----|---------------------------------------|-----|
| ly:font-load.....                      | 203 | ly:make-music.....                    | 206 |
| ly:font-magnification.....             | 203 | ly:make-music-function.....           | 207 |
| ly:font-metric?.....                   | 203 | ly:make-output-def.....               | 207 |
| ly:font-name.....                      | 203 | ly:make-page-label-marker.....        | 207 |
| ly:font-sub-fonts.....                 | 203 | ly:make-page-permission-marker.....   | 207 |
| ly:format.....                         | 203 | ly:make-pango-description-string..... | 207 |
| ly:format-output.....                  | 203 | ly:make-paper-outputter.....          | 207 |
| ly:get-all-function-documentation..... | 203 | ly:make-pitch.....                    | 207 |
| ly:get-all-translators.....            | 203 | ly:make-prob.....                     | 207 |
| ly:get-glyph.....                      | 203 | ly:make-scale.....                    | 207 |
| ly:get-listened-event-classes.....     | 203 | ly:make-score.....                    | 207 |
| ly:get-option.....                     | 203 | ly:make-simple-closure.....           | 207 |
| ly:gettext.....                        | 203 | ly:make-stencil.....                  | 207 |
| ly:grob-alist-chain.....               | 203 | ly:make-stream-event.....             | 207 |
| ly:grob-array-length.....              | 203 | ly:message.....                       | 207 |
| ly:grob-array-ref.....                 | 204 | ly:minimal-breaking.....              | 207 |
| ly:grob-array?.....                    | 204 | ly:mm.....                            | 208 |
| ly:grob-basic-properties.....          | 204 | ly:module->alist.....                 | 208 |
| ly:grob-common-refpoint.....           | 204 | ly:module-copy.....                   | 208 |
| ly:grob-common-refpoint-of-array.....  | 204 | ly:modules-lookup.....                | 208 |
| ly:grob-default-font.....              | 204 | ly:moment-add.....                    | 208 |
| ly:grob-extent.....                    | 204 | ly:moment-div.....                    | 208 |
| ly:grob-interfaces.....                | 204 | ly:moment-grace-denominator.....      | 208 |
| ly:grob-layout.....                    | 204 | ly:moment-grace-numerator.....        | 208 |
| ly:grob-object.....                    | 204 | ly:moment-main-denominator.....       | 208 |
| ly:grob-original.....                  | 204 | ly:moment-main-numerator.....         | 208 |
| ly:grob-parent.....                    | 204 | ly:moment-mod.....                    | 208 |
| ly:grob-pq<?.....                      | 204 | ly:moment-mul.....                    | 208 |
| ly:grob-properties.....                | 204 | ly:moment-sub.....                    | 208 |
| ly:grob-property.....                  | 204 | ly:moment<?.....                      | 208 |
| ly:grob-property-data.....             | 204 | ly:moment?.....                       | 208 |
| ly:grob-relative-coordinate.....       | 204 | ly:music-compress.....                | 208 |
| ly:grob-robust-relative-extent.....    | 204 | ly:music-deep-copy.....               | 208 |
| ly:grob-script-priority-less.....      | 204 | ly:music-duration-compress.....       | 208 |
| ly:grob-set-property!.....             | 205 | ly:music-duration-length.....         | 208 |
| ly:grob-staff-position.....            | 205 | ly:music-function-extract.....        | 209 |
| ly:grob-suicide!.....                  | 205 | ly:music-function?.....               | 209 |
| ly:grob-system.....                    | 205 | ly:music-length.....                  | 209 |
| ly:grob-translate-axis!.....           | 205 | ly:music-list?.....                   | 209 |
| ly:grob?.....                          | 205 | ly:music-mutable-properties.....      | 209 |
| ly:gulp-file.....                      | 205 | ly:music-output?.....                 | 209 |
| ly:hash-table-keys.....                | 205 | ly:music-property.....                | 209 |
| ly:inch.....                           | 205 | ly:music-set-property!.....           | 209 |
| ly:input-both-locations.....           | 205 | ly:music-transpose.....               | 209 |
| ly:input-file-line-char-column.....    | 205 | ly:music?.....                        | 209 |
| ly:input-location?.....                | 205 | ly:note-head::stem-attachment.....    | 209 |
| ly:input-message.....                  | 205 | ly:number->string.....                | 209 |
| ly:interpret-music-expression.....     | 205 | ly:optimal-breaking.....              | 209 |
| ly:interpret-stencil-expression.....   | 205 | ly:option-usage.....                  | 209 |
| ly:intlog2.....                        | 205 | ly:otf->cff.....                      | 209 |
| ly:is-listened-event-class.....        | 205 | ly:otf-font-glyph-info.....           | 209 |
| ly:item-break-dir.....                 | 205 | ly:otf-font-table-data.....           | 209 |
| ly:item?.....                          | 206 | ly:otf-font?.....                     | 209 |
| ly:iterator?.....                      | 206 | ly:otf-glyph-list.....                | 210 |
| ly:lexer-keywords.....                 | 206 | ly:output-def-clone.....              | 210 |
| ly:lily-lexer?.....                    | 206 | ly:output-def-lookup.....             | 210 |
| ly:lily-parser?.....                   | 206 | ly:output-def-parent.....             | 210 |
| ly:load-text-dimensions.....           | 206 | ly:output-def-scope.....              | 210 |
| ly:make-book.....                      | 206 | ly:output-def-set-variable!.....      | 210 |
| ly:make-dispatcher.....                | 206 | ly:output-def?.....                   | 210 |
| ly:make-duration.....                  | 206 | ly:output-description.....            | 210 |
| ly:make-global-context.....            | 206 | ly:output-formats.....                | 210 |
| ly:make-global-translator.....         | 206 | ly:outputter-close.....               | 210 |
| ly:make-listener.....                  | 206 | ly:outputter-dump-stencil.....        | 210 |
| ly:make-moment.....                    | 206 | ly:outputter-dump-string.....         | 210 |

|                                       |     |
|---------------------------------------|-----|
| ly:outputter-output-scheme.....       | 210 |
| ly:outputter-port .....               | 210 |
| ly:page-marker?.....                  | 210 |
| ly:page-turn-breaking .....           | 210 |
| ly:pango-font-physical-fonts .....    | 210 |
| ly:pango-font?.....                   | 210 |
| ly:paper-book-pages.....              | 210 |
| ly:paper-book-paper.....              | 211 |
| ly:paper-book-performances.....       | 211 |
| ly:paper-book-scopes .....            | 211 |
| ly:paper-book-systems .....           | 211 |
| ly:paper-book?.....                   | 211 |
| ly:paper-fonts .....                  | 211 |
| ly:paper-get-font .....               | 211 |
| ly:paper-get-number.....              | 211 |
| ly:paper-outputscale .....            | 211 |
| ly:paper-score-paper-systems .....    | 211 |
| ly:paper-system-minimum-distance..... | 211 |
| ly:paper-system?.....                 | 211 |
| ly:parse-file.....                    | 211 |
| ly:parser-clear-error .....           | 211 |
| ly:parser-clone .....                 | 211 |
| ly:parser-define!.....                | 211 |
| ly:parser-error .....                 | 211 |
| ly:parser-has-error? .....            | 211 |
| ly:parser-lexer .....                 | 211 |
| ly:parser-lookup.....                 | 212 |
| ly:parser-output-name .....           | 212 |
| ly:parser-parse-string .....          | 212 |
| ly:parser-set-note-names.....         | 212 |
| ly:performance-write .....            | 212 |
| ly:pfb->pfa.....                      | 212 |
| ly:pitch-alteration.....              | 212 |
| ly:pitch-diff.....                    | 212 |
| ly:pitch-negate .....                 | 212 |
| ly:pitch-notename .....               | 212 |
| ly:pitch-octave .....                 | 212 |
| ly:pitch-quartertines .....           | 212 |
| ly:pitch-semitones.....               | 212 |
| ly:pitch-steps .....                  | 212 |
| ly:pitch-transpose.....               | 212 |
| ly:pitch<?.....                       | 212 |
| ly:pitch? .....                       | 212 |
| ly:position-on-line? .....            | 212 |
| ly:prob-immutable-properties .....    | 212 |
| ly:prob-mutable-properties.....       | 213 |
| ly:prob-property .....                | 213 |
| ly:prob-property? .....               | 213 |
| ly:prob-set-property! .....           | 213 |
| ly:prob-type?.....                    | 213 |
| ly:prob? .....                        | 213 |
| ly:programming-error .....            | 213 |
| ly:progress.....                      | 213 |
| ly:property-lookup-stats .....        | 213 |
| ly:protects.....                      | 213 |
| ly:pt.....                            | 213 |
| ly:register-stencil-expression.....   | 213 |
| ly:relative-group-extent.....         | 213 |
| ly:reset-all-fonts.....               | 213 |
| ly:round-filled-box.....              | 213 |
| ly:round-filled-polygon.....          | 213 |
| ly:run-translator .....               | 213 |
| ly:score-add-output-def! .....        | 214 |
| ly:score-embedded-format .....        | 214 |
| ly:score-error? .....                 | 214 |

|                                           |     |
|-------------------------------------------|-----|
| ly:score-header .....                     | 214 |
| ly:score-music .....                      | 214 |
| ly:score-output-defs .....                | 214 |
| ly:score-set-header! .....                | 214 |
| ly:score? .....                           | 214 |
| ly:set-default-scale .....                | 214 |
| ly:set-grob-modification-callback .....   | 214 |
| ly:set-middle-C! .....                    | 214 |
| ly:set-option .....                       | 214 |
| ly:set-point-and-click .....              | 214 |
| ly:set-property-cache-callback .....      | 214 |
| ly:simple-closure? .....                  | 214 |
| ly:skyline-pair? .....                    | 214 |
| ly:skyline? .....                         | 215 |
| ly:smob-protects .....                    | 215 |
| ly:solve-spring-rod-problem .....         | 215 |
| ly:source-file? .....                     | 215 |
| ly:spanner-bound .....                    | 215 |
| ly:spanner-broken-into .....              | 215 |
| ly:spanner? .....                         | 215 |
| ly:start-environment .....                | 215 |
| ly:stderr-redirect .....                  | 215 |
| ly:stencil-add .....                      | 215 |
| ly:stencil-aligned-to .....               | 215 |
| ly:stencil-combine-at-edge .....          | 215 |
| ly:stencil-empty? .....                   | 215 |
| ly:stencil-expr .....                     | 215 |
| ly:stencil-extent .....                   | 215 |
| ly:stencil-fonts .....                    | 216 |
| ly:stencil-in-color .....                 | 216 |
| ly:stencil-rotate .....                   | 216 |
| ly:stencil-translate .....                | 216 |
| ly:stencil-translate-axis .....           | 216 |
| ly:stencil? .....                         | 216 |
| ly:stream-event? .....                    | 216 |
| ly:string-substitute .....                | 216 |
| ly:system-print .....                     | 216 |
| ly:system-stretch .....                   | 216 |
| ly:text-dimension .....                   | 216 |
| ly:text-interface::interpret-markup ..... | 216 |
| ly:translator-description .....           | 216 |
| ly:translator-group? .....                | 216 |
| ly:translator-name .....                  | 216 |
| ly:translator? .....                      | 216 |
| ly:transpose-key-alist .....              | 216 |
| ly:truncate-list! .....                   | 217 |
| ly:ttf->pfa .....                         | 217 |
| ly:ttf-ps-name .....                      | 217 |
| ly:unit .....                             | 217 |
| ly:usage .....                            | 217 |
| ly:version .....                          | 217 |
| ly:warning .....                          | 217 |
| ly:wide-char->utf-8 .....                 | 217 |
| Lyrics .....                              | 177 |

## M

|                              |          |
|------------------------------|----------|
| m .....                      | 79       |
| magnifying text .....        | 138      |
| maj .....                    | 79       |
| majorSevenSymbol .....       | 82       |
| makeClusters .....           | 113, 197 |
| manuelle Systemwechsel ..... | 61       |
| Marcato .....                | 30, 172  |

|                                |            |
|--------------------------------|------------|
| Measure_grouping_engraver      | 19         |
| MeasureGrouping                | 19         |
| Mehrstimmigkeit                | 45         |
| Mensur                         | 91         |
| Mensural_ligature_engraver     | 87, 95, 96 |
| Mensuralligaturen              | 95         |
| MensuralStaffContext           | 102        |
| MensuralVoiceContext           | 102        |
| merging text                   | 145        |
| Metrum                         | 18         |
| Mezzosopranschlüssel           | 8          |
| MIDI-Instrumentenbezeichnungen | 118        |
| Mikrotöne                      | 4          |
| minimumFret                    | 66         |
| Modi, in Akkorden              | 79         |
| Modifying context properties   | 74         |
| Modus                          | 9          |
| Mordent                        | 30, 172    |
| Musica ficta                   | 104        |
| musicMap                       | 113, 197   |

## N

|                                        |               |
|----------------------------------------|---------------|
| N-tolen                                | 12            |
| N-tolen, Formatierung                  | 13            |
| Niente, al                             | 32            |
| Non-guitar tablatures                  | 67            |
| noPageBreak                            | 113, 198      |
| noPageTurn                             | 113, 198      |
| note-collision-interface               | 187, 189, 191 |
| note-event                             | 71            |
| Note_heads_engraver                    | 21            |
| NoteCollision                          | 49, 51        |
| NoteColumn                             | 50            |
| NoteHead                               | 5, 86         |
| Notenbezeichnungen, andere Sprachen    | 5             |
| Notenbezeichnungen, Deutsch            | 3             |
| Notenbezeichnungen, Standard           | 3             |
| Notenkopfarten                         | 134           |
| Notenköpfe, Alte Musik                 | 86            |
| Notenköpfe, Stile                      | 46            |
| Notenlänge                             | 11            |
| Notenlinien, Anzahl                    | 55            |
| Notenlinien, Dicke                     | 55            |
| Notenschlüssel                         | 8             |
| Notensysteme, mehrere                  | 53            |
| Notensysteme, Modifikation             | 55            |
| notes within text by log and dot-count | 162           |
| notes within text by string            | 162           |

## O

|                                          |          |
|------------------------------------------|----------|
| octaveCheck                              | 113, 198 |
| Offen                                    | 30, 172  |
| Oktavenüberprüfung                       | 6        |
| oldaddlyrics                             | 114, 198 |
| Orgelpedalbezeichnung                    | 30, 172  |
| Ossia                                    | 55       |
| ottava                                   | 114, 198 |
| overrideProperty                         | 114, 198 |
| overriding properties within text markup | 169      |

## P

|                                         |          |
|-----------------------------------------|----------|
| padding text                            | 152      |
| padding text horizontally               | 152      |
| pageBreak                               | 114, 198 |
| pageTurn                                | 114, 198 |
| Parallele Notation, Eingabe             | 51       |
| parallelMusic                           | 114, 198 |
| parenthesize                            | 114, 198 |
| partcombine                             | 114, 198 |
| Partitur                                | 53       |
| Pausen                                  | 17       |
| Pausen, Alte Musik                      | 88       |
| Pedalbezeichnung                        | 63       |
| PercentRepeat                           | 43       |
| PercentRepeatCounter                    | 43       |
| PercentRepeatedMusic                    | 43       |
| Percussion                              | 71       |
| Perkussion                              | 71       |
| Phrasierungsbögen                       | 35       |
| PhrasingSlur                            | 35       |
| Piano templates                         | 61       |
| PianoPedalBracket                       | 64       |
| PianoStaff                              | 38       |
| pipeSymbol                              | 25       |
| Pitch_squash_engraver                   | 180      |
| pitchedTrill                            | 114, 198 |
| placing horizontal brackets around text | 159      |
| placing vertical brackets around text   | 157      |
| Platzhalternoten                        | 17       |
| pointAndClickOff                        | 114, 198 |
| pointAndClickOn                         | 114, 198 |
| Polyphonie                              | 45       |
| Portato                                 | 30, 172  |
| Praller                                 | 30, 172  |
| Prallermordent                          | 30, 172  |
| Prima volta                             | 40       |
| Prozent-Wiederholungen                  | 43       |
| Punktierung                             | 11       |
| putting space around text               | 152      |

## Q

|                              |          |
|------------------------------|----------|
| Quadratische Neumenligaturen | 96       |
| quoteDuring                  | 114, 198 |

## R

|                                    |          |
|------------------------------------|----------|
| r                                  | 17       |
| raising text                       | 153      |
| Rechte Hand-Fingersatz für Gitarre | 69       |
| referencing page numbers in text   | 170      |
| Relativ                            | 2        |
| Relative Oktavbestimmung           | 2        |
| removeWithTag                      | 114, 198 |
| repeatCommands                     | 25       |
| repeatCommands                     | 42       |
| RepeatedMusic                      | 42       |
| RepeatedMusic                      | 43       |
| RepeatSlash                        | 43       |
| resetRelativeOctave                | 114, 198 |
| Rest                               | 17, 88   |
| RestCollision                      | 51       |
| RhythmicStaff                      | 22       |
| Rhythmische Aufteilungen           | 12       |

right aligning text..... 153  
 rightHandFinger ..... 114, 198  
 rotating text ..... 154

## S

**s** ..... 17  
 scaleDurations..... 114, 198  
 scaling text ..... 154  
 Schlagzeug..... 71  
 Schlüssel..... 8  
 Schlüssel, Alte Musik..... 88  
 Score ..... 19, 174, 176  
 scoreTweak ..... 114, 198  
 Script ..... 31  
 Seconda volta ..... 40  
 Segno ..... 30, 172  
 setting extent of text objects ..... 171  
 setting horizontal text alignment ..... 147  
 setting subscript in standard font size ..... 138  
 setting superscript in standard font size ..... 139  
 shiftDurations..... 114, 198  
 simple text strings ..... 140  
 simple text strings with tie characters ..... 165  
 Single staff ..... 62  
 Skip ..... 17  
 SkipMusic ..... 18  
 slashed digits ..... 170  
 Slur ..... 35  
 Sopranschlüssel ..... 8  
 spacing-spanner-interface..... 194  
 spacingTweaks..... 114, 198  
 SpanBar ..... 25  
 Staccatissimo..... 30, 172  
 Staccato ..... 30, 172  
 stacking text in a column ..... 144  
 Staff ..... 18, 25, 93, 174  
 StaffGroup ..... 53  
 StaffSymbol ..... 55  
 Stem ..... 57  
 Stem ..... 90  
 stemLeftBeamCount ..... 23  
 stemRightBeamCount ..... 23  
 StemTremolo ..... 44  
 Stile, Notenköpfe..... 134  
 Stimmgruppe ..... 53  
 Stimmkreuzung ..... 62  
 storePredefinedDiagram ..... 114, 199  
 String numbers ..... 65  
 StringNumber ..... 66  
 StrokeFinger ..... 70  
 Subbassschlüssel ..... 8  
 subdivideBeams ..... 23  
 subscript text..... 141  
 Subtraktion in Akkorden..... 78  
 suggestAccidentals..... 104  
 superscript text..... 142  
 sus ..... 79  
 SustainPedal ..... 64  
 Symbole, Akkord-..... 80  
 System querende Hälse ..... 63  
 System, Chor ..... 53  
 Systeme, mehrere ..... 53  
 Systemgruppe..... 53

SystemStartBar ..... 54  
 SystemStartBrace ..... 54  
 SystemStartBracket ..... 54  
 Systemwechsel von Stimmen..... 62  
 Systemwechsel, manuell ..... 61

## T

Tab\_note\_heads\_engraver ..... 68  
 Tablatures basic ..... 66  
 TabStaff ..... 66, 67  
 TabVoice ..... 66, 67  
 tag ..... 114, 199  
 Taktangabe ..... 18  
 Taktart, Alte Musik..... 91  
 Takte verkürzen ..... 19  
 Taktlinien..... 24  
 Taktüberprüfung..... 25  
 Taktweise Wiederholungen ..... 43  
 Tenorschlüssel..... 8  
 Tenorschlüssel, Chor ..... 8  
 Tenuto ..... 30, 172  
 text columns, left-aligned ..... 151  
 text columns, right-aligned ..... 153  
 Text formatieren..... 191, 195  
 text-interface ..... 169  
 TextScript ..... 31  
 Tie ..... 16  
 TimeScaledMusic ..... 14  
 TimeSignature..... 19  
 TimeSignature ..... 91  
 Timing\_translator ..... 19, 176  
 tocItem..... 114, 199  
 Tonart ..... 9  
 Tonhöhenbezeichnungen..... 1  
 translating text ..... 154  
 Transponieren..... 7  
 transposedCueDuring..... 115, 199  
 TransposedMusic ..... 7  
 transposition..... 115, 199  
 Transposition ..... 7  
 Tremolobalken ..... 44  
 tremoloFlags ..... 44  
 Tremolozeichen ..... 44  
 Triller ..... 30, 38, 172  
 Triller mit Tonhöhe ..... 39  
 TrillSpanner..... 39  
 Triolen..... 12  
 TupletBracket..... 14  
 TupletNumber ..... 14  
 tupletNumberFormatFunction..... 13  
 tweak ..... 115, 199

## U

Überbindung ..... 15  
 Überspringen von Zeichen ..... 17  
 underlining text..... 143  
 UnfoldedRepeatedMusic ..... 42  
 UnfoldedRepeatedMusic ..... 43  
 unfoldRepeats ..... 115, 199  
 Unsichtbare Pausen ..... 17

**V**

|                                        |                    |
|----------------------------------------|--------------------|
| Varcoda .....                          | 30, 172            |
| Vaticana_ligature_engraver .....       | 87                 |
| VaticanaStaffContext .....             | 102                |
| VaticanaVoiceContext .....             | 102                |
| Verschachtelte Musik .....             | 51                 |
| Versetzungszeichen .....               | 3                  |
| Versetzungszeichen, Erinnerung .....   | 4                  |
| Versetzungszeichen, musica ficta ..... | 104                |
| Versetzungszeichen, Vierteltöne .....  | 4                  |
| Versetzungszeichen, Warnung .....      | 4                  |
| vertically centering text .....        | 155                |
| Verzierungen .....                     | 26                 |
| Vierteltöne .....                      | 4                  |
| Violinschlüssel .....                  | 8                  |
| Voice .....                            | 18, 32, 46, 47, 95 |
| VoiceFollower .....                    | 63                 |
| Volta .....                            | 40                 |
| Volta und Überbindung .....            | 15                 |
| Volta_engraver .....                   | 80                 |
| VoltaBracket .....                     | 42                 |
| VoltaBracket .....                     | 43                 |
| VoltaRepeatedMusic .....               | 42                 |
| VoltaRepeatedMusic .....               | 43                 |
| Vorhalt .....                          | 26                 |
| Vorschlag .....                        | 26                 |
| Vorzeichen .....                       | 9                  |

|                               |    |
|-------------------------------|----|
| Vorzeichen in Klammern .....  | 4  |
| Vorzeichen, Alte Musik .....  | 87 |
| Vorzeichen, Deutsch .....     | 3  |
| Vorzeichen, Erinnerung .....  | 4  |
| Vorzeichen, Vierteltöne ..... | 4  |

**W**

|                                    |          |
|------------------------------------|----------|
| Warnungsvorzeichen .....           | 4        |
| Wechsel des Systems, manuell ..... | 61       |
| Wechsel zwischen Systemen .....    | 62       |
| Weißer Mensuralligaturen .....     | 95       |
| whichBar .....                     | 25       |
| Wiederholung und Überbindung ..... | 15       |
| Wiederholung, mehrdeutig .....     | 42       |
| Wiederholungen .....               | 39       |
| Wiederholungszeichen .....         | 24       |
| withMusicProperty .....            | 115, 199 |

**Z**

|                                       |    |
|---------------------------------------|----|
| Zahl der Notenlinien einstellen ..... | 55 |
| Zeichen .....                         | 29 |
| Zeitberechnung, intern .....          | 26 |
| Ziernoten .....                       | 26 |
| Zweite Klammer .....                  | 40 |