Das Notensatzprogramm

Notationsreferenz
Das LilyPond-Entwicklerteam

Dieses Handbuch stellt eine Referenz aller Notationsformen zur Verfügung, die mit LilyPond Version 2.19.84 erstellt werden können. Es wird vorausgesetzt, dass der Leser mit dem Abschnitt “Handbuch zum Lernen” in Handbuch zum Lernen vertraut ist.

Zu mehr Information, wie dieses Handbuch unter den anderen Handbüchern positioniert, oder um dieses Handbuch in einem anderen Format zu lesen, besuchen Sie bitte Abschnitt “Manuals” in Allgemeine Information.

Wenn Ihnen Handbücher fehlen, finden Sie die gesamte Dokumentation unter http://lilypond.org/.

Copyright © 1999–2015 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.


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”.

Für LilyPond Version 2.19.84
Inhaltsverzeichnis

1 Musikalische Notation .............................................. 1
  1.1 Tonhöhen .................................................................. 1
      1.1.1 Tonhöhen setzen ................................................ 1
             Absolute Oktavenbezeichnung ................................ 1
             Relative Oktavenbezeichnung .................................. 2
             Versetzungssymbole ............................................. 5
             Notenbezeichnungen in anderen Sprachen ..................... 7
      1.1.2 Viele Tonhöhen gleichzeitig verändern ...................... 9
             Oktavenüberprüfung ............................................. 9
             Transponieren .................................................... 10
             Umkehrung ....................................................... 13
             Krebs ............................................................. 14
             Modale Transformierungen .................................... 14
      1.1.3 Tonhöhen anzeigen lassen .................................... 16
             Notenschlüssel .................................................... 16
             Tonartbezeichnung ............................................... 20
             Oktavierungsromane ............................................. 22
             Transposition von Instrumenten ................................ 23
             Automatische Versetzungssymbole ............................. 24
             Tonumfang ......................................................... 31
      1.1.4 Notenköpfchen .................................................. 34
             Besondere Notenköpfchen ....................................... 34
             Easy-Notation-Notenköpfe ........................................ 35
             Notenköpfe mit besonderen Formen ............................ 37
             Improvisation ..................................................... 40
  1.2 Rhythmus ............................................................... 40
      1.2.1 Rhythmen eingeben ............................................. 41
             Tondauern ......................................................... 41
             Andere rhythmische Aufteilungen ................................. 43
             Tondauern skalieren .............................................. 48
             Bindebogen ....................................................... 49
      1.2.2 Pausen eingeben ................................................ 52
             Pausen ........................................................... 52
             Unsichtbare Pausen ............................................... 54
             Ganztaktpausen .................................................... 56
      1.2.3 Rhythmen anzeigen lassen .................................... 60
             Taktangabe ........................................................ 60
             Metronomangabe ................................................ 65
             Auftakte ........................................................ 68
             Musik ohne Metrum .............................................. 69
             Polymetrische Notation ......................................... 71
             Automatische Aufteilung von Noten ............................ 74
             Melodierhythmus anzeigen ..................................... 75
      1.2.4 Balken ............................................................. 77
             Automatische Balken ............................................. 77
             Einstellung von automatischen Balken ......................... 80
             Manuelle Balken ............................................... 88
             Gespreizte Balken ............................................... 91
<table>
<thead>
<tr>
<th>Seite</th>
<th>Kapitel</th>
<th>Unterkapitel</th>
<th>Titel</th>
<th>Seitenzahl</th>
</tr>
</thead>
<tbody>
<tr>
<td>191</td>
<td>1.7</td>
<td>1.7.1</td>
<td>Innerhalb des Systems</td>
<td></td>
</tr>
<tr>
<td>198</td>
<td>1.7</td>
<td>1.7.2</td>
<td>Außenhalb des Notensystems</td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>1.7</td>
<td>Anmerkungen</td>
<td></td>
<td>203</td>
</tr>
<tr>
<td>204</td>
<td>1.7.1</td>
<td>Auswahl der Notations-Schriftgröße</td>
<td>204</td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>1.7.1</td>
<td>Fingersatzanweisungen</td>
<td>205</td>
<td></td>
</tr>
<tr>
<td>207</td>
<td>1.7.1</td>
<td>Unsichtbare Noten</td>
<td>207</td>
<td></td>
</tr>
<tr>
<td>208</td>
<td>1.7.1</td>
<td>Farbige Objekte</td>
<td>208</td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>1.7.1</td>
<td>Klamern</td>
<td>210</td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>1.7.1</td>
<td>Hülse</td>
<td>210</td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>1.7.2</td>
<td>Erklärungen in Ballonform</td>
<td>211</td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>1.7.2</td>
<td>Gitternetzlinien</td>
<td>212</td>
<td></td>
</tr>
<tr>
<td>214</td>
<td>1.7.2</td>
<td>Analyseklammern</td>
<td>214</td>
<td></td>
</tr>
<tr>
<td>215</td>
<td>1.8</td>
<td>Text</td>
<td>215</td>
<td></td>
</tr>
<tr>
<td>216</td>
<td>1.8.1</td>
<td>Text eingeben</td>
<td>216</td>
<td></td>
</tr>
<tr>
<td>216</td>
<td>1.8.1</td>
<td>Textarten</td>
<td>216</td>
<td></td>
</tr>
<tr>
<td>217</td>
<td>1.8.1</td>
<td>Text mit Verbindungslinien</td>
<td>217</td>
<td></td>
</tr>
<tr>
<td>219</td>
<td>1.8.1</td>
<td>Textartige Zeichen</td>
<td>219</td>
<td></td>
</tr>
<tr>
<td>222</td>
<td>1.8.1</td>
<td>Separater Text</td>
<td>222</td>
<td></td>
</tr>
<tr>
<td>223</td>
<td>1.8.2</td>
<td>Text formatieren</td>
<td>223</td>
<td></td>
</tr>
<tr>
<td>223</td>
<td>1.8.2</td>
<td>Textbeschriftung (Einleitung)</td>
<td>223</td>
<td></td>
</tr>
<tr>
<td>225</td>
<td>1.8.2</td>
<td>Überblick über die wichtigsten Textbeschriftungsbefehle</td>
<td>225</td>
<td></td>
</tr>
<tr>
<td>228</td>
<td>1.8.2</td>
<td>Textausrichtung</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>231</td>
<td>1.8.2</td>
<td>Graphische Notation innerhalb einer Textbeschriftung</td>
<td>231</td>
<td></td>
</tr>
<tr>
<td>234</td>
<td>1.8.2</td>
<td>Musikalische Notation innerhalb einer Textbeschriftung</td>
<td>234</td>
<td></td>
</tr>
<tr>
<td>236</td>
<td>1.8.2</td>
<td>Textbeschriftung über mehrere Seiten</td>
<td>236</td>
<td></td>
</tr>
<tr>
<td>237</td>
<td>1.8.3</td>
<td>Schriftarten</td>
<td>237</td>
<td></td>
</tr>
<tr>
<td>237</td>
<td>1.8.3</td>
<td>Was sind Schriftarten</td>
<td>237</td>
<td></td>
</tr>
<tr>
<td>239</td>
<td>1.8.3</td>
<td>Schriftarten für einen Eintrag</td>
<td>239</td>
<td></td>
</tr>
<tr>
<td>239</td>
<td>1.8.3</td>
<td>Schriftart des gesamten Dokuments</td>
<td>239</td>
<td></td>
</tr>
</tbody>
</table>

2 Spezielle Notation ................................................. 241

2.1 Notation von Gesang ............................................. 241

2.1.1 Übliche Notation für Vokalmusik ................................ 241

Referenz für Vokalmusik ........................................... 241

Eingabe von Text .................................................. 242

Text an einer Melodie ausrichten .................................. 244

Automatische Silbendauern ....................................... 245

Manuelle Silbendauern ........................................... 248

Mehrere Silben zu einer Note .................................... 249

Mehrere Noten zu einer Silbe .................................... 250

Fülllinien und Trennstriche ..................................... 253

2.1.2 Techniken für die Gesangstextnotation ....................... 254

Mit Gesangstexten und Bezeichnern arbeiten ..................... 254

Gesangstext vertikal verschieben ................................ 255

Silben horizontal verschieben .................................. 260

Gesangstext und Wiederholungen ................................ 262

Getrennte Texte .................................................. 270

2.1.3 Strophen ..................................................... 271

Strophennummern hinzufügen .................................... 271

Lautstärkebezeichnung zu Strophen hinzufügen .................. 271

Sängernamen zu Strophen hinzufügen ............................. 272
2.2 Tasteninstrumente und andere Instrumente mit mehreren Systemen

2.3 Bundlose Saiteninstrumente

2.1.4 Lieder

2.1.5 Chormusik

2.1.6 Oper und Musical

2.2.1 Übliche Notation für Tasteninstrumente

2.2.2 Klavier

2.2.3 Akkordeon

2.2.4 Harfe

2.3.1 Übliche Notation für bundlose Saiteninstrumente

2.4.1 Übliche Notation für Saiteninstrumente mit Bünden

2.4.2 Gitarre
Flageolett und gedämpfte Noten .................................................. 361
Powerakkorde anzeigen .............................................................. 363
2.4.3 Banjo ................................................................. 364
Banjo-Tabulaturen ................................................................. 364
2.5 Schlagzeug ................................................................. 365
2.5.1 Übliche Notation für Schlagzeug ........................................ 365
Referenz für Schlagzeug ............................................................. 365
Grundlagen der Schlagzeugnotation ............................................ 365
Trommelwirbel ................................................................. 366
Schlagzeug mit Tonhöhe ............................................................. 367
Schlagzeugsysteme ................................................................. 367
Eigene Schlagzeugsysteme ......................................................... 369
Geisternoten ................................................................. 373
2.6 Blasinstrumente ........................................................... 374
2.6.1 Übliche Notation für Bläser ................................................ 374
Referenz für Blasinstrumente ...................................................... 374
Fingersatz ................................................................. 375
2.6.2 Dudelsack ............................................................. 377
Dudelsack-Definitionen ........................................................... 377
Dudelsack-Beispiele ............................................................... 378
2.6.3 Holzbläser ............................................................. 379
2.6.3.1 Holzbläserdiagramme .............................................. 379
2.7 Notation von Akkorden .................................................... 387
2.7.1 Akkord-Modus .......................................................... 388
Überblick über den Akkord-Modus ............................................ 388
Übliche Akkorde ................................................................. 389
Erweiterte und modifizierte Akkorde .......................................... 390
2.7.2 Akkorde anzeigen ........................................................ 393
Akkordbezeichnungen drucken .................................................. 393
Akkordbezeichnungen anpassen ............................................... 396
2.7.3 Generalbass ........................................................... 401
Grundlagen des Bezifferten Basses ............................................ 402
Eingabe des Generalbass’ ......................................................... 402
Generalbass anzeigen .............................................................. 405
2.8 Zeitgenössische Musik ...................................................... 408
2.8.1 Tonhöhe und Harmonie in zeitgenössischer Musik ................. 408
Verweise zu Tonhöhe und Harmonie in zeitgenössischer Musik .... 408
Mikrotonale Notation ............................................................ 408
Zeitgenössische Tonartvorzeichnung und Harmonie .................... 409
2.8.2 Zeitgenössische Notation von Rhythmen .......................... 409
Verweise für zeitgenössische Benutzung von Rhythmus .............. 409
N-tolen in zeitgenössischer Musik .............................................. 409
Zeitgenössische Takttarten ....................................................... 409
Erweiterte polymetrische Notation ............................................. 409
Balken in zeitgenössischer Musik .............................................. 409
Taktstriche in zeitgenössischer Musik ......................................... 409
2.8.3 Graphische Notation .................................................. 409
2.8.4 Zeitgenössische Partiturtechniken ................................... 409
2.8.5 Neue Instrumententechniken .......................................... 409
2.8.6 Leseliste und interessante Referenzpartituren ..................... 409
Bücher und Artikel über zeitgenössische Notation ...................... 409
Partituren und Musikbeispiele .................................................. 409
2.9 Notation von alter Musik ................................................ 409
2.9.1 Überblick über die unterstützten Stile .............................. 411
3 Allgemeine Eingabe und Ausgabe 447

3.1 Eingabestruktur 447
3.1.1 Struktur einer Partitur 447
3.1.2 Mehrere Partituren in einem Buch 448
3.1.3 Mehrere Ausgabedateien aus einer Eingabedatei 449
3.1.4 Dateinamen der Ausgabedateien ........................................... 450
3.1.5 Die Dateistruktur ................................................................. 451
3.2 Titel ......................................................................................... 453
  3.2.1 Titel, Kopf- und Fußzeilen erstellen ......................................... 453
  Wie funktioniert die Titel-Umgebung? ............................................. 453
  Standardlayout von book- und Partitur-Titelumgebungen .................. 456
  Standardlayout von Kopf- und Fußzeilen ........................................ 459
  3.2.2 Eigene Kopf- und Fußzeilen sowie Titel .................................... 460
  Angepasste Textformatierung für Titelumgebungen ......................... 460
  Angepasstes Layout für Titelumgebungen ....................................... 461
  Angepasstes Layout für Kopf- und Fußzeilen ................................... 464
  3.2.3 Fußnoten erstellen ................................................................. 465
    Übersicht über Fußnoten ............................................................ 466
    Automatische Fußnoten ............................................................. 466
    Manuelle Fußnoten ................................................................. 469
  3.2.4 Verweis auf die Seitenzahlen ................................................ 472
  3.2.5 Inhaltsverzeichnis ............................................................... 474
3.3 Arbeiten an Eingabe-Dateien .................................................. 476
  3.3.1 LilyPond-Dateien einfügen .................................................. 476
  3.3.2 Verschiedene Editionen aus einer Quelldatei ............................. 477
    Variablen benutzen ................................................................. 477
    Marken benutzen .................................................................. 479
    Globale Einstellungen benutzen ................................................. 482
  3.3.3 sonderzeichen ................................................................. 482
    Zeichenkodierung ................................................................. 482
    Unicode ................................................................................ 483
    ASCII-Alias .......................................................................... 484
3.4 Ausgabe kontrollieren ............................................................ 485
  3.4.1 Notationsfragmente extrahieren .............................................. 485
  3.4.2 Korrigierte Musik überspringen .............................................. 485
  3.4.3 Alternative Ausgabeformate ................................................ 486
  3.4.4 Die Notationsschriftart verändern ........................................ 486
3.5 MIDI-Ausgabe ................................................................. 487
  3.5.1 MIDI-Dateien erstellen ........................................................ 487
    Instrumentenbezeichnungen ....................................................... 488
  3.5.2 Der MIDI-Block ................................................................. 489
  3.5.3 Was geht in die MIDI-Ausgabe .............................................. 490
    In MIDI unterstützt ............................................................... 490
    In MIDI nicht unterstützt ...................................................... 490
  3.5.4 Wiederholungen im MIDI .................................................... 491
  3.5.5 MIDI-Lautstärke kontrollieren .............................................. 491
    Dynamik-Zeichen ................................................................. 491
    MIDI-Lautstärke .................................................................. 492
    Verschiedene Instrumente angleichen (i) .................................. 493
    Verschiedene Instrumente angleichen (ii) .................................. 494
  3.5.6 Schlagzeug in MIDI ............................................................. 495
  3.5.7 Artikulier-Skript ................................................................. 496
3.6 Musikalische Information extrahieren ....................................... 496
  3.6.1 LilyPond-Notation anzeigen ................................................ 496
  3.6.2 Musikalische Scheme-Ausdrücke anzeigen ................................ 497
  3.6.3 Musikalische Ereignisse in einer Datei speichern ..................... 497
4 Abstände ............................................. 498
4.1 Seitenlayout ........................................... 498
  4.1.1 Die \paper-Umgebung ................................. 498
  4.1.2 Papierformat und automatische Skalierung ............... 499
    Das Papierformat einstellen ................................ 499
    Automatische Skalierung auf ein Papierformat ............... 500
  4.1.3 Vertikale \paper-Variablen mit festen Abständen .......... 500
  4.1.4 Vertikale \paper-Variablen mit flexiblen Abständen ........ 501
    Struktur der Alisten für flexible vertikale Abstände .......... 501
    Liste der flexiblen vertikalen Abstandsvariablen in \paper .. 502
  4.1.5 \paper-Variablen für horizontale Abstände ............... 503
    \paper-Variablen für Breite und Ränder ...................... 504
    \paper-Variablen für zwei seitigen Satz ...................... 505
    \paper-Variablen für Verschiebungen und Einrückungen ....... 505
  4.1.6 Andere \paper-Variablen .............................. 506
    \paper-Variablen für den Zeilenumbruch ...................... 506
    \paper-Variablen für den Seitenumbruch ...................... 506
    \paper-Variablen für Seitenzahlen ........................... 507
    Verschiedene \paper-Variablen ................................ 508
4.2 Partiturlayout ......................................... 509
  4.2.1 Die \layout-Umgebung ................................. 509
  4.2.2 Die Notensystemgröße einstellen ....................... 511
4.3 Umbrüche .............................................. 512
  4.3.1 Zeilenumbrüche ...................................... 512
  4.3.2 Seitenumbrüche ...................................... 514
  4.3.3 Optimale Seitenumbrüche ................................ 515
  4.3.4 Optimale Umbrüche zum Blättern ....................... 515
  4.3.5 Minimale Seitenumbrüche ................................ 516
  4.3.6 Eine-Seite-Seitenumbrüche ............................. 517
  4.3.7 Ausdrückliche Umbrüche ................................ 517
  4.3.8 Eine zusätzliche Stimme für Umbrüche benutzen .......... 518
4.4 Vertikale Abstände ..................................... 520
  4.4.1 Flexible vertikale Abstände in Systemgruppen ............ 520
    Eigenschaften für Abstände innerhalb von Systemgruppen ...... 521
    Abstände von nicht gruppierten Notensystemen ............... 524
    Abstände von gruppierten Notensystemen ...................... 525
    Abstände von nicht-Notensystemzeilen ....................... 527
  4.4.2 Explizite Positionierung von Systemen .................. 528
  4.4.3 Vermeidung von vertikalen Zusammenstößen ............... 535
4.5 Horizontale Abstände ................................... 536
  4.5.1 Überblick über horizontale Abstände .................... 536
  4.5.2 Eine neuer Bereich mit anderen Abständen ............... 538
  4.5.3 Horizontale Abstände verändern ....................... 538
  4.5.4 Zeilenlänge .......................................... 541
  4.5.5 Proportionale Notation ................................ 541
4.6 Die Musik auf weniger Seiten zwingen ....................... 548
  4.6.1 Abstände anzeigen lassen ................................ 549
  4.6.2 Abstände verändern .................................... 550
5 Standardeinstellungen verändern ............................................. 552
  5.1 Interpretationskontexte ......................................................... 552
    5.1.1 Was sind Kontexte? .................................................. 552
    5.1.2 Kontexte erstellen und referenzieren ............................. 554
    5.1.3 Kontexte am Leben halten ......................................... 555
    5.1.4 Umgebungs-Plugins verändern ..................................... 558
    5.1.5 Die Standardeinstellungen von Kontexten verändern .......... 560
      Alle Kontexte des gleichen Typs verändern ......................... 560
      Nur einen bestimmten Kontext verändern ......................... 563
      Rangfolge von Kontextwerten ...................................... 565
    5.1.6 Neue Kontexte definieren ........................................ 565
    5.1.7 Reihenfolge des Kontextlayouts .................................. 567
  5.2 Die Referenz der Programminterna erklärt ........................... 568
    5.2.1 Zurechtfinden in der Programminterna ......................... 568
    5.2.2 Layout-Schmittstellen ........................................... 569
    5.2.3 Die Grob-Eigenschaften ......................................... 571
    5.2.4 Benennungskonventionen ...................................... 571
    5.2.5 Mittlere Kontexte – Systeme ................................... 572
    5.2.6 Oberste Kontexte – Container für Systeme ..................... 572
    5.2.7 Untere Kontexte – Stimmen ..................................... 572
    5.2.8 Score – der Vater aller Kontexte ................................. 572
    5.3 Eigenschaften verändern .............................................. 572
      5.3.1 Grundlagen zum Verändern von Eigenschaften ................ 572
      5.3.2 Der \set-Befehl .................................................. 572
      5.3.3 Der \override-Befehl .......................................... 574
      5.3.4 Der \tweak-Befehl .............................................. 576
      5.3.5 \set versus \override .......................................... 578
      5.3.6 Alisten verändern ............................................. 578
  5.4 Nützliche Konzepte und Eigenschaften ................................ 580
    5.4.1 Eingabe-Modi ...................................................... 580
    5.4.2 Richtung und Platzierung ....................................... 581
    5.4.3 Abstände und Maße .................................................. 583
    5.4.4 Eigenschaften des Staff-Symbols ................................ 583
    5.4.5 Strecker  ................................................................ 584
      Das spanner-interface benutzen ..................................... 584
      Das line-spanner-interface benutzen ................................ 587
    5.4.6 Sichtbarkeit von Objekten ........................................ 589
      Einen stencil entfernen .............................................. 589
      Objekte unsichtbar machen ......................................... 590
      Objekte weiß malen .................................................. 590
      \break-visibility (unsichtbar machen) benutzen .................. 590
      Besonderheiten ...................................................... 592
    5.4.7 Linienstile ............................................................ 594
    5.4.8 Drehen von Objekten .............................................. 595
      Drehen von Layout-Objekten ....................................... 595
      Textbeschriftung drehen ........................................... 596
  5.5 Fortgeschrittene Optimierungen .......................................... 596
    5.5.1 Objekte ausrichten .................................................. 596
      X-offset und Y-offset direkt setzen ................................ 597
      Das side-position-interface benutzen ................................ 597
      Das self-alignment-interface benutzen ................................ 598
      Benutzung des break-alignable-interface .......................... 599
    5.5.2 Vertikale Gruppierung der grafischen Objekte („grob“s) ..... 601
    5.5.3 stencils verändern ................................................. 601
Anhang A Notationsübersicht ........................................................................ 608
A.1 Liste der Akkordbezeichnungen ....................................................... 608
A.2 Übliche Akkord-Variablen ................................................................. 609
A.3 Vordefinierte Saitenstimmungen ..................................................... 612
A.4 Die vordefinierten Bund-Diagramme ................................................. 613
A.5 Vordefinierte Papierformate ............................................................. 619
A.6 MIDI-Instrumente .............................................................................. 622
A.7 Liste der Farben ................................................................................... 623
A.8 Die Emmentaler-Schriftart ................................................................. 625
   Notenschlüssel-Glyphen ................................................................. 625
   Taktart-Glyphen ............................................................................... 625
   Zahlen-Glyphen ............................................................................... 626
   Versetzungsglypchen ...................................................................... 626
   Standard-Notenkopf-Glyphen ......................................................... 627
   Spezielle Notenkopf-Glyphen ......................................................... 627
   Geformte Notenkopf-Glyphen ......................................................... 628
   Pausen-Glyphen ............................................................................... 632
   Fähnchen-Glyphen .......................................................................... 633
   Punkt-Glyphen ................................................................................. 633
   Dynamik-Glyphen ............................................................................ 633
   Schrift-Glyphen ............................................................................... 634
   Pfeilkopf-Glyphen ............................................................................ 636
   Klammer spitzen-Glyphen ............................................................... 636
   Pedal-Glyphen .................................................................................. 636
   Akkordeon-Glyphen ........................................................................ 637
   Bogen-Glyphen ................................................................................ 637
   Vaticana-Glyphen ............................................................................ 637
   Medicamaa-Glyphen ........................................................................ 638
   Hufnagel-Glyphen ............................................................................ 639
   Mensural-Glyphen ............................................................................ 639
   Neomensural-Glyphen ..................................................................... 643
   Petrucci-Glyphen .............................................................................. 644
   Solesmes-Glyphen ............................................................................ 645
   Glyphen der Kiever Notation ............................................................ 645
A.9 Notenkopfstile .................................................................................... 646
A.10 Textbeschriftungsbefehle ................................................................. 646
   A.10.1 Font ......................................................................................... 646
   A.10.2 Align ....................................................................................... 656
   A.10.3 Graphic ................................................................................... 672
   A.10.4 Music ...................................................................................... 680
   A.10.5 Instrument Specific Markup ................................................... 686
   A.10.6 Accordion Registers ............................................................... 689
   A.10.7 Other ....................................................................................... 694
A.11 Textbeschriftungslistenbefehle .......................................................... 701
A.12 Liste der Sonderzeichen ................................................................. 703
A.13 Liste der Artikulationszeichen .......................................................... 705
Artikulationsscripte ................................................................................. 705
Ornamentale Skripte .......................................................... 705
Fermatenskripte .............................................................. 705
Instrumentenspezifische Skripte ........................................... 705
Wiederholungszeichenskripte .............................................. 706
Ancient scripts ............................................................... 706
A.14 Schlagzeugnoten .......................................................... 706
A.15 Technisches Glossar ..................................................... 708
alist ........................................................... 708
callback .......................................................... 708
closure .......................................................... 708
glyph .......................................................... 708
grob .......................................................... 708
immutable .......................................................... 709
interface .......................................................... 709
lexer .......................................................... 709
mutable .......................................................... 709
output-def .......................................................... 709
parser .......................................................... 710
parser variable ......................................................... 710
prob .......................................................... 710
simple closure ........................................................ 710
smob .......................................................... 710
stencil .......................................................... 711
A.16 Alle Kontexteigenschaften ............................................. 711
A.17 Eigenschaften des Layouts ........................................... 724
A.18 Erhältliche Musikfunktionen ......................................... 745
A.19 Bezeichner zur Kontextveränderung ................................. 756
A.20 Vordefinierte Typprädikate ........................................... 756
R5RS primary predicates ............................................... 756
R5RS secondary predicates ............................................ 757
Guile predicates ......................................................... 757
LilyPond scheme predicates ............................................ 757
LilyPond exported predicates ........................................ 758
A.21 Scheme-Funktionen ..................................................... 759

Anhang B  Befehlsübersicht ................................................. 785
Anhang C  GNU Free Documentation License ........................... 789
Anhang D  Index der LilyPond-Befehle .................................... 796
Anhang E  LilyPond-Index .................................................... 804
1 Musikalische Notation

Dieses Kapitel erklärt, wie die Notation von Musik erstellt wird.

1.1 Tonhöhen

Dieser Abschnitt zeigt, wie man die Tonhöhe notieren kann. Es gibt drei Stufen in diesem Prozess: Eingabe, Veränderung und Ausgabe.

1.1.1 Tonhöhen setzen


Absolute Oktavenbezeichnung


```
\clef bass
\c4 \d \e \f
\g4 \a \b \c
\d4 \e \f \g
```

Durch die Verwendung von deutscher Notenbezeichnungen mit der Unterscheidung von b und h kann man die Tonhöhe genauer bestimmen. Die Notenbezeichnungen c bis b werden in der Oktave unter dem eingestrichenen C gesetzt.
Andere Oktaven können erreicht werden, indem man ein Apostroph (') oder ein Komma (,) benutzt. Jedes ' erhöht die Tonhöhe um eine Oktave, jedes , erniedrigt sie um eine Oktave.

```
clef treble
c'4 c' ' e' g
d''4 d' d c
clef bass
c,4 c,, e, g
d,,4 d, d c
```

Siehe auch
- Glossar: Abschnitt “Pitch names” in Glossar.
- Schnipsel: Abschnitt “Pitches” in Schnipsel.

Relative Oktavenbezeichnung

Wenn Oktaven im absoluten Modus notiert, passiert es schnell, eine Note auf der falschen Oktave zu notieren. Mit dem relativen Modus kommen solche Fehler seltener vor, weil man die Oktave nur noch sehr selten spezifizieren muss. Hinzu kommt, dass im absoluten Modus ein einzelner Fehler schwer zu finden ist, während er im relativen Modus den ganzen Rest des Stückes um eine Oktave verschiebt.

```
relative Anfangstonhöhe musikalischer Ausdruck
```

Im relativen Modus wird angenommen, dass sich jede folgende Note so dicht wie möglich bei der nächsten befindet. Das bedeutet, dass die Oktave jeder Tonhöhe innerhalb eines musikalischen Ausdrucks wie folgt errechnet wird:

- Wenn kein Oktavänderungszeichen an einer Tonhöhe benutzt wird, wird ihre Oktave so errechnet, dass das Intervall zur vorigen Note weniger als eine Quinte ist. Das Intervall wird errechnet, ohne Versetzungszeichen zu berücksichtigen.
- Ein Oktavänderungszeichen ' oder , kann hinzugefügt werden, um eine Tonhöhe explizit um eine Oktave zu erhöhen bzw. zu erniedrigen, relativ zu der Tonhöhe, die ohne das Oktavänderungszeichen errechnet wurde.
- Mehrfache Oktavänderungszeichen können benutzt werden. Die Zeichen'' und ,, ändern zum Beispiel die Tonhöhe um zwei Oktaven.
- Die Tonhöhe der ersten Note ist relativ zu Anfangstonhöhe. Die Anfangstonhöhe wird im absoluten Modus gesetzt, und als Empfehlung gilt, eine Oktave von C zu nehmen.

So funktioniert der relative Modus:

```
relative {
  clef bass
c d e f
g a b c
d e f g
}
```
Oktavversetzungen müssen für alle Intervalle angezeigt werden, die größer als eine Quarte sind.
\[ \begin{aligned} \text{relative} & \{ \\
& c'' g c f, \\
& c' a, e'' c \} \end{aligned} \]

Eine Sequenz ohne ein einziges Oktavänderungszeichen kann aber trotzdem weite Intervalle umfassen:
\[ \begin{aligned} \text{relative} & \{ \\
& c f b e \\
& a d g c \} \end{aligned} \]

Wenn \texttt{relative}-Umgebungen geschachtelt werden, gilt der innerste \texttt{relative}-Abschnitt.
\[ \begin{aligned} \text{relative} & \{ \\
& c' d e f \\
& \text{relative} \{ \\
& c'' d e f \} \\
& \} \} \end{aligned} \]

\texttt{relative} hat keine Auswirkung auf \texttt{chordmode}-Abschnitte.
\[ \begin{aligned} \text{new Staff} & \{ \\
& \text{relative} c'' \{ \\
& \text{chordmode} \{ c1 \} \}\\
& \} \end{aligned} \]

\texttt{relative} darf nicht innerhalb von \texttt{chordmode} notiert werden.
Kapitel 1: Musikalische Notation

Tonhöhen innerhalb eines \texttt{transpose}-Abschnitts sind absolut, es sei denn ein \texttt{relative} wird eingefügt.

\begin{verbatim}
\relative { 
d' e 
\transpose f g { 
d e 
\relative { 
d' e 
}
}
}
\end{verbatim}

Wenn der vorherige Ausdruck ein Akkord ist, wird die erste Note des Akkordes benutzt, um die erste Note des nächsten Akkordes zu bestimmen. Innerhalb von Akkorden ist die nächste Note immer relativ zur vorherigen. Betrachten Sie das folgende Beispiel aufmerksam, insbesondere die c-Noten.

\begin{verbatim}
\relative { 
c' <c e g> <c' e g'> <c, e, g''>
}
\end{verbatim}

Wie oben erklärt wurde, wird die Oktave einer Tonhöhe nur nach ihrer Notenbezeichnung errechnet, unabhängig von allen Versetzungszeichen. Darum wird ein Eisis auf ein H (notiert als b) folgend höher gesetzt, während ein Feses tiefer gesetzt wird. Anders gesagt wird eine doppelterhöhte Quarte wird als kleineres Intervall angesehen als eine doppeltvölkerniedrigte Quinte, unabhängig von der Anzahl an Halbtönen, die jedes Intervall enthält.

\begin{verbatim}
\relative { 
c'''2 fis c2 ges b2 eisis b2 feses
}
\end{verbatim}

Eine Konsequenz dieser Regeln ist, dass die erste Note innerhalb von \texttt{relative f} auf die selbe Art interpretiert wird als wenn sie im absoluten Tonhöhenmodus geschrieben worden wäre.
Siehe auch


Notationsreferenz: [Oktavenüberprüfung], Seite 9.
Schnipsel: Abschnitt “Pitches” in Schnipsel.

Wenn keine Anfangstonhöhe für \relative angegeben wird, wird c' angenommen. Das ist aber eine veraltete Option, die in späteren Programmversionen verschwinden kann. Darum wird von der Benutzung abgeraten.

Versetzungszeichen


\relative c'' { ais1 aes aisis aees }

Auch die deutschen Varianten as für aes und es für ees sind erlaubt. Im Unterschied zum Deutschen ist aber bes die einzige Version für den Ton B, während his als bis geschrieben werden muss. Das kann aber auch verändert werden, siehe [Notenbezeichnungen in anderen Sprachen], Seite 7.

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

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 eingestrichenen C.

\relative c'' { ceseh1 ces ceh c cih cis cisih }
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

Versetzungszeichen von übergebundenen Noten werden nur dann gesetzt, wenn ein neues System begonnen wird:
cis\! cis\!
\break
cis

Ausgewählte Schnipsel

*Hiding accidentals on tied notes at the start of a new system*

This shows how to hide accidentals on tied notes at the start of a new system.

\relative c' {
\override Accidental.hide-tied-accidental-after-break = ##t
cis\! cis\!
\break
cis
}

Verhindern, dass zusätzliche Auflösungszeichen automatisch hinzugefügt werden

Den traditionellen Notensatzregeln zufolge wird ein Auflösungszeichen immer dann vor einem Kreuz oder B gesetzt, wenn ein vorheriges Versetzungszeichen der gleichen Note aufgehoben
werden soll. Um dieses Verhalten zu ändern, muss die Eigenschaft `extraNatural` im Staff-Kontext auf "false" gesetzt werden.

```lilypond
\relative c' { 
    aeses4 aes ais a 
    \set Staff.extraNatural = ##f 
    aeses4 aes ais a 
}
```

Siehe auch


Handbuch zum Lernen: Abschnitt “Tonhöhen und Tonartbezeichnungen (Vorzeichen)” in Handbuch zum Lernen.

Notationsreferenz: [Automatische Versetzungszeichen], Seite 24, [Vorgeschlagene Versetzungszeichen (musica ficta)], Seite 419, [Notenbezeichnungen in anderen Sprachen], Seite 7.

Schnipsel: Abschnitt “Pitches” in Schnipsel.


**Bekannte Probleme und Warnungen**

Es gibt keine allgemeinen Regeln für die Notation von Vierteltönen, die Symbole von LilyPond folgen also keinem Standard.

**Notenbezeichnungen in anderen Sprachen**

Es gibt vordefinierte Bezeichnungen für die Notenbezeichnungen in anderen Sprachen als Englisch. Die Sprache für die Notenbezeichnungen wird normalerweise zu Beginn einer Datei ausgewählt: das folgende Beispiel zeigt die Verwendung von italienischen Notenbezeichnungen:

```lilypond
\language "italiano"

\relative { 
    do' re mi sib 
}
```

In der Tabelle sind die existierenden Sprachdefinitionen mit den dazugehörigen Notenbezeichnungen dargestellt.

<table>
<thead>
<tr>
<th>Sprache</th>
<th>Notenbezeichnungen</th>
</tr>
</thead>
<tbody>
<tr>
<td>nederlands</td>
<td>c d e f g a bes b</td>
</tr>
<tr>
<td>catalan</td>
<td>do re mi fa sol la sib si</td>
</tr>
</tbody>
</table>
Kapitel 1: Musikalische Notation

und die dazugehörigen Versetzungszeichen-Endungen:

<table>
<thead>
<tr>
<th>Sprache</th>
<th>Kreuz</th>
<th>B</th>
<th>Doppelkreuz</th>
<th>Doppel-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>nederlands</td>
<td>-ih</td>
<td>-eh</td>
<td>-isih</td>
<td>-eseh</td>
</tr>
<tr>
<td>catalan</td>
<td>-ih</td>
<td>-eh</td>
<td>-isih</td>
<td>-eseh</td>
</tr>
<tr>
<td>deutsch</td>
<td>-ih</td>
<td>-eh</td>
<td>-isih</td>
<td>-eseh</td>
</tr>
<tr>
<td>english</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
<td>-tqf</td>
</tr>
<tr>
<td>espanol</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
<td>-tqf</td>
</tr>
<tr>
<td>italiano</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
<td>-tqf</td>
</tr>
<tr>
<td>norsk</td>
<td>-iss</td>
<td>-iss</td>
<td>-ississ</td>
<td>-eses</td>
</tr>
<tr>
<td>suomi</td>
<td>-iss</td>
<td>-iss</td>
<td>-ississ</td>
<td>-eses</td>
</tr>
<tr>
<td>portugues</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
<td>-tqf</td>
</tr>
<tr>
<td>svenska</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
<td>-tqf</td>
</tr>
<tr>
<td>vlaams</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
<td>-tqf</td>
</tr>
</tbody>
</table>


\relative c'' { a2 as e es a a ses e eses }


<table>
<thead>
<tr>
<th>Sprache</th>
<th>Vierteltonkreuz</th>
<th>Viertelton-3/4-Tonkreuz</th>
<th>3/4-Tonkreuz</th>
</tr>
</thead>
<tbody>
<tr>
<td>nederlands</td>
<td>-ih</td>
<td>-eh</td>
<td>-isih</td>
</tr>
<tr>
<td>deutsch</td>
<td>-ih</td>
<td>-eh</td>
<td>-isih</td>
</tr>
<tr>
<td>english</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
</tr>
<tr>
<td>espanol</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
</tr>
<tr>
<td>italiano</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
</tr>
<tr>
<td>portugues</td>
<td>-qs</td>
<td>-qf</td>
<td>-tqs</td>
</tr>
</tbody>
</table>

Die meisten Sprachen, die hier vorkommen, werden normalerweise mit der klassischen westlichen Musik assoziiert. Alternative Tonhöhen und Stimmungen sind aber auch unterstützt: siehe Abschnitt 2.10.1 [Übliche Notation für nichteuropäische Musik], Seite 439,
Siehe auch

Glossar: Abschnitt “Pitch names” in Glossar, Abschnitt “Common Practice Period” in Glossar.

Notationsreferenz: Abschnitt 2.10.1 [Übliche Notation für nichteuropäische Musik], Seite 439.
Installierte Dateien: scm/define-note-names.scm.
Schnipsel: Abschnitt “Pitches” in Schnipsel.

1.1.2 Viele Tonhöhen gleichzeitig verändern
Dieser Abschnitt zeigt, wie man Tonhöhen beeinflusst.

Oktavenüberprüfung
Im relativen Modus geschieht es recht häufig, dass ein Oktavänderungszeichen vergessen wird. Oktavenüberprüfungen machen es einfacher, solche Fehler zu entdecken und zu korrigieren. Sie geben eine Warnung aus und korrigieren die Oktave, wenn eine Note in einer unerwarteten Oktave gefunden wird.

Um die Oktave einer Note zu überprüfen, muss die absolute Oktave nach dem =-Symbol angegeben werden. Im folgenden Beispiel wird eine Warnung (und ein Oktavänderung) generiert, weil die zweite Note als absolute Oktave ein d'' anstelle von d' notiert ist, wie es die Oktavierungskorrektur markiert.

\relative {
  c''2 d''4 d
  e2 f
}

\relative {
  \octaveCheck c'
  e2 f
}

Die Oktave von einer Note kann auch mit dem \octaveCheck Kontrolltonhöhe-Befehl überprüft werden. Kontrolltonhöhe wird im absoluten Modus eingegeben. Dabei wird überprüft, ob das Intervall zwischen der vorherigen Note und der Kontrolltonhöhe nicht größer als eine Quarte ist (die normale Berechnung im relativen Modus). Wenn diese Überprüfung einen Fehler ausgibt, wird eine Warnung gemeldet, aber die vorigen Note wird nicht verändert. Folgende Noten sind dann relativ zur Kontrolltonhöhe.

\relative {
  c''2 d
  \octaveCheck c'
  e2 f
}

Vergleichen Sie die zwei Takte im nächsten Beispiel. Die erste und dritte \octaveCheck-Überprüfung gibt einen Fehler aus, die zweite dagegen ist erfolgreich:

\relative {
  c''4 f g f
  c4
Transponieren

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

\transpose vonTonhöhe nachTonhöhe mus. Ausdruck

Das bedeutet, dass der mus. Ausdruck um das Intervall zwischen den Tonhöhen vonTonhöhe und nachTonhöhe transponiert wird: Jede Note, die die Tonhöhe vonTonhöhe hat, wird in die Tonhöhe nachTonhöhe umgewandelt, und alle anderen Noten um das gleiche Intervall. Beide Tonhöhen werden im absoluten Modus eingegeben.

Achtung: Tonhöhen innerhalb einer \transpose-Umgebung sind absolut, es sei denn, ein \relative wird eingefügt.

So kann z. B. ein Stück in D-Dur, wenn es für den Sänger etwas zu tief ist, nach E-Dur transponiert werden. Dabei werden auch die Vorzeichen entsprechend angepasst:

\transpose d e {
    \relative {
        \key d \major
        d'4 fis a d
    }
}

WENN eine Stimme, die in C notiert ist, von einer A-Klarinette gespielt werden soll (für die A als C notiert wird, aber eine kleine Terz tiefer erklingt als es notiert ist), kann die entsprechende Stimme wie folgt erstellt werden:

\transpose a c' {
    \relative {
        \key c \major
        c'4 d e g
    }
}
Beachten Sie, dass \key c \major explizit angegeben werden muss. Wenn hier keine Tonart angemerkt würde, würde die Noten zwar transponiert, aber keine Vorzeichen angezeigt werden.

\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 Versetzungsschreiben Kreuze aus, die zweite dagegen B-Versetzungsschreiben.

\transpose kann auch benutzt werden, um die geschriebenen Noten eines transponierenden Instruments zu notieren. Im vorigen Beispiel wurde die Tonhöhen so eingegeben, wie sie erklingen (also in C), aber man kann genauso gut auch andersherum aus einer Stimme, die für ein transponierendes Instrument in einem anderen Ton als C geschrieben wurde, eine Partitur in C erstellen. Die Noten einer B-Trompete, die mit einem notierten E (also einem klingenden D) anfangen, könnte man also auch so eingeben:

\transpose c bes, \musicInBflat

Um die Noten dann in F zu setzen (um sie etwa für ein Horn zu arrangieren), könnte man die schon geschriebenen Noten wieder mit einem weiteren \transpose umgeben:

\transpose f c' \transpose c bes, \musicInBflat

Für mehr Information zu transponierenden Instrumenten siehe auch [Transposition von Instrumenten], Seite 23.

**Ausgewählte Schnipsel**

*Noten mit minimaler Anzahl an Versetzungsschreiben transponieren.*

Dieses Beispiel benutzt Scheme-Code, um enharmonische Verwechslungen für Noten zu erzwingen, damit nur eine minimale Anzahl an Versetzungsschreiben ausgegeben wird. In diesem Fall gelten die folgenden Regeln:

- Doppelte Versetzungsschreiben sollen entfernt werden
- His -> C
- Eis -> F
- Ces -> B
- Fes -> E

Auf diese Art werden am meisten natürliche Tonhöhen als enharmonische Variante gewählt.

#(define (naturalize-pitch p)
Kapitel 1: Musikalische Notation

(let ((o (ly:pitch-octave p))
     (a (* 4 (ly:pitch-alteration p)))
     ;; alteration, a, in quarter tone steps,
     ;; for historical reasons
     (n (ly:pitch-notename p)))

(cond
  ((and (> a 1) (or (eqv? n 6) (eqv? n 2)))
   (set! a (- a 2))
   (set! n (+ n 1)))
  ((and (< a -1) (or (eqv? n 0) (eqv? n 3)))
   (set! a (+ a 2))
   (set! n (- n 1)))
  ((> a 2) (set! a (- a 4)) (set! n (+ n 1)))
  (< a -2) (set! a (+ a 4)) (set! n (- n 1)))

(if (< n 0) (begin (set! o (- o 1)) (set! n (+ n 7))))
(if (> n 6) (begin (set! o (+ o 1)) (set! n (- n 7))))
(ly:make-pitch o n (/ a 4)))

#(define (naturalize music)
   (let ((es (ly:music-property music 'elements))
         (e (ly:music-property music 'element))
         (p (ly:music-property music 'pitch)))
     (if (pair? es)
         (ly:music-set-property!
          music 'elements
          (map naturalize es)))
     (if (ly:music? e)
         (ly:music-set-property!
          music 'element
          (naturalize e)))
     (if (ly:pitch? p)
         (begin
          (set! p (naturalize-pitch p))
          (ly:music-set-property! music 'pitch p)))
     music))

naturalizeMusic =
#(define-music-function (m)
   (ly:music?)
   (naturalize m))

music = 'relative c' { c4 d e g }

\score {
  \new Staff {
    \transpose c ais { \music }
    \naturalizeMusic \transpose c ais { \music }
    \transpose c deses { \music }
    \naturalizeMusic \transpose c deses { \music }
  }
  \layout { }
Kapitel 1: Musikalische Notation

Siehe auch

Notationsreferenz: [Relative Oktavenbezeichnung], Seite 2, [Transposition von Instrumenten], Seite 23. [Umkehrung], Seite 13, [Krebs], Seite 14, [Modale Transformierungen], Seite 14.

Schnipsel: Abschnitt “Pitches” in Schnipsel.


Bekannte Probleme und Warnungen

Der relative Modus wirkt nicht in \transpose, \chordmode oder \relative. Um auch im relativen Modus transponieren zu können, muss ein \relative innerhalb des \transpose zusätzlich gesetzt werden.

Dreifache Versetzungssymbole werden nicht ausgegeben, wenn \transpose benutzt wird. Eine enharmonisch entsprechende Tonhöhe wird anstelle dessen gesetzt (z. B. Des anstelle von e-dreifach-b).

Umkehrung

Ein musikalischer Ausdruck kann in seine Umkehrung transformiert und gleichzeitig transponiert werden:

\inversion umTonhöhe nachTonhöhe mus. Ausdruck

Der mus. Ausdruck wird Intervall für Intervall um vUmTonhöhe umgekehrt und dann von vUmTonhöhe nach nachTonhöhe transponiert.

\begin{verbatim}
music = \relative { c' d e f }
\new Staff {
  \music
  \inversion d' d' \music
  \inversion d' ees' \music
}\end{verbatim}

\begin{verbatim}
\end{verbatim}

Achtung: Motive, die umgekehrt werden, sollen in absoluten Tonhöhen notiert werden oder zuerst in absolute Tonhöhen umgewandelt werden, indem sie in eine \relative-Umgebung eingeschlossen werden.

Siehe auch

Notationsreferenz: [Modale Transformierungen], Seite 14, [Krebs], Seite 14, [Transponieren], Seite 10.
Krebs

Ein musikalischer Ausdruck kann umgekehrt werden, um im Krebsgang dargestellt zu werden:

\begin{verbatim}
music = \relative { c'8. ees16( fis8. a16 b8.) gis16 f8. d16 }
\end{verbatim}

\begin{verbatim}
\new Staff {
  \music
  \retrograde \music
}\end{verbatim}

Bekannte Probleme und Warnungen

Manuell gesetzte Bindebögen innerhalb des Krebsgangs \texttt{retrograde} werden aufgebrochen und erzeugen Warnungen. Einige Bindebögen können automatisch ersetzt werden, indem man ⟨\texttt{undefined}⟩ \texttt{Automatic note splitting}, Seite ⟨\texttt{undefined}⟩, aktiviert.

Siehe auch

Notationsreferenz: [Modale Transformierungen], Seite 14, [Umkehrung], Seite 13, [Transponieren], Seite 10.

Modale Transformierungen

In einer Komposition, die auf einer Skala basiert, wird ein Motiv oft in unterschiedlicher Weise transformiert. Es kann transponiert werden, um von einer anderen Tonhöhe in der Skala zu beginnen, oder beginnend von einer bestimmten Tonhöhe der Skala umgekehrt werden. Es kann auch im Krebsgang, also rückwärts erscheinen, siehe [Krebs], Seite 14.

\begin{verbatim}
\textbf{Achtung:} Alle Noten, die nicht zu der vorgegebenen Skala gehören, werden nicht transformiert.
\end{verbatim}

Modale Transposition

Ein Motiv kann innerhalb einer Skala transponiert werden:

\begin{verbatim}
\texttt{modalTranspose vonTonhöhe nachTonhöhe Skala mus. Ausdruck}
\end{verbatim}

Die Noten von \texttt{mus. Ausdruck} werden innerhalb von \texttt{Skala} um die Anzahl von Skalenstufen transponiert, die dem Intervall zwischen \texttt{vonTonhöhe} und \texttt{nachTonhöhe} entspricht.

\begin{verbatim}
diatonicScale = \relative { c' d e f g a b }
motif = \relative { c'8 d e f g a b c }
\end{verbatim}

\begin{verbatim}
\new Staff {
  \motif
  \modalTranspose c f \diatonicScale \motif
  \modalTranspose c b, \\diatonicScale \motif
}\end{verbatim}
Eine absteigende Skala beliebiger Länge und mit beliebigen Intervallen kann definiert werden:

```
pentatonicScale = \relative { ges aes bes des ees }
motif = \relative { ees'8 des,4 <ges' bes,> <ges bes,> }
```

\new Staff {
\motif
\modalTranspose ges ees' \pentatonicScale \motif
}

Wenn `\modalTranspose` mit einer chromatischen Skala benutzt wird, entspricht es der Funktion von `\transpose` mit dem Unterschied, dass die Bezeichnungen der eingesetzten Noten definiert werden können:

```
chromaticScale = \relative { c' cis d dis e f gis a ais b }
motif = \relative { c'8 d e f g a b c }
```

\new Staff {
\motif
\transpose c f \motif
\modalTranspose c f \chromaticScale \motif
}

**Modale Umkehrung**

Ein musikalischer Ausdruck kann innerhalb einer definierten Skala um eine Referenznote umgekehrt werden und transponiert werden:

```
\modalInversion Umkehrungstonhöhe nachTonhöhe Skala mus. Ausdruck
```


Um also einfach eine Umkehrung ohne zusätzliche Transposition zu machen, sollten `Umkehrungstonhöhe` und `nachTonhöhe` gleich sein.

```
octatonicScale = \relative { ees' f fis gis a b c d }
motif = \relative { c'8. ees16 fis8. a16 b8. gis16 f8. d16 }
```

\new Staff {
\motif
\modalInversion fis' fis' \octatonicScale \motif
}
Um den Drehpunkt der Umkehrung zwischen zwei Noten der Skala zu definieren, wird um eine der Noten die Umkehrung vorgenommen und dann eine Skalenstufe transponiert. Die beiden Noten klammern sozusagen den Drehpunkt ein.

\begin{verbatim}
scale = \relative { c' g' }
motive = \relative { c' c g' c, }

\new Staff {
  \motive
  \modalInversion c' g' \scale \motive
}
\end{verbatim}

Die kombinierte Operation von Umkehrung und Krebs erzeugt die Krebsumkehrung:

\begin{verbatim}
octatonicScale = \relative { ees' f fis gis a b c d }
motif = \relative { c'8. ees16 fis8. a16 b8. gis16 f8. d16 }

\new Staff {
  \motive
  \retrograde \modalInversion c' c' \octatonicScale \motive
}
\end{verbatim}

Siehe auch

Notationsreferenz: [Umkehrung], Seite 13, [Krebs], Seite 14, [Transponieren], Seite 10.

1.1.3 Tonhöhen anzeigen lassen
Dieser Abschnitt zeigt, wie die Ausgabe von Tonhöhen verändert werden kann.

Notenschlüssel
Der Schlüssel kann verändert werden. Das eingestrichene C wird in jedem Beispiel gezeigt:

\begin{verbatim}
\clef treble
c'2 c'
\clef alto
c'2 c'
\clef tenor
c'2 c'
\clef bass
c'2 c'
\end{verbatim}
Andere Schlüssel sind u. A.:
\clef french
c2 c
\clef soprano
c2 c
\clef mezzosoprano
c2 c
\clef baritone
c2 c
\break
\clef varbaritone
c2 c
\clef subbass
c2 c
\clef percussion
c2 c
\break
\clef G % synonym for treble
c2 c
\clef F % synonym for bass
c2 c
\clef C % synonym for alto
c2 c

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. Auch andere Ganzzahlen können verwendet werden, wenn es gewünscht wird. Die Schlüsselbezeichnung muss in Anführungszeichen gesetzt werden, wenn nicht-alphabetische Zeichen enthalten, siehe Beispiel:
\clef treble
c'2 c'
\clef "treble_8"
c'2 c'
\clef "bass^15"
c'2 c'
Kapitel 1: Musikalische Notation

Genau wie in [Men surale Schlüssel], Seite 414, [Gregorianische Schlüssel], Seite 422, [Standardtabulaturen], Seite 322, und [Angepasste Tabulaturen], Seite 335, sind verschiedene Schlüssel unterstützt. Zur Benutzung unterschiedlicher Schlüssel bei Stichnoten siehe die Funktionen \cufClef und \cufDuringWithClef in Ausgewählte Schnipsel

**Eigenschaften des Schlüssels optimieren**

Der Befehl \cufClef "treble_8" ist gleichbedeutend mit einem expliziten Setzen der Eigenschaften von clefGlyph, clefPosition (welche die vertikale Position des Schlüssels bestimmt), middleCPosition und clefTransposition. Ein Schlüssel wird ausgegeben, wenn eine der Eigenschaften außer middleCPosition sich ändert.


Wenn ein Schlüsselwechsel an einem Zeilenwechsel geschieht, wird das neue Symbol sowohl am Ende der alten Zeilen als auch am Anfang der neuen Zeile ausgegeben. Wenn der Warnungs-Schlüssel am Ende der alten Zeile nicht erforderlich ist, kann er unterdrückt werden, indem die explicitClefVisibility-Eigenschaft des Staff-Kontextes auf den Wert end-of-line-invisible gesetzt wird. Das Standardverhalten kann mit \unset Staff.explicitClefVisibility wieder hergestellt werden.

Die folgenden Beispiele zeigen die Möglichkeiten, wenn man diese Eigenschaften manuell setzt. Auf der ersten Zeile erhalten die manuellen Änderungen die ursprüngliche relative Positionierung von Schlüssel und Noten, auf der zweiten Zeile nicht.

\{% The default treble clef
\key f \major
  c'1
%
% The standard bass clef
\set Staff.clefGlyph = "clefs.F"
\set Staff.clefPosition = #2
\set Staff.middleCPosition = #6
\set Staff.middleCClefPosition = #6
\key g \major
  c'1
%
% The baritone clef
\set Staff.clefGlyph = "clefs.C"
\key f \major
  c'2
\set Staff.clefGlyph = "clefs.F"
\set Staff.clefPosition = #2
\set Staff.middleCPosition = #6
\set Staff.middleCClefPosition = #6
\key g \major
  c'1
% The baritone clef
\set Staff.clefGlyph = "clefs.C"
Kapitel 1: Musikalische Notation

\set Staff.clefPosition = #4
\set Staff.middleCPosition = #4
\set Staff.middleCClefPosition = #4
\key f \major
\c'\break
% The standard choral tenor clef
\set Staff.clefGlyph = "clefs.G"
\set Staff.clefPosition = #-2
\set Staff.clefTransposition = #-7
\set Staff.middleCPosition = #1
\set Staff.middleCClefPosition = #1
\key f \major
\c'\break
% A non-standard clef
\set Staff.clefGlyph = "clefs.F"
\set Staff.clefPosition = #0
\set Staff.clefTransposition = #0
\set Staff.middleCPosition = #-4
\set Staff.middleCClefPosition = #-4
\key g \major
\c'\break
% The following clef changes do not preserve
% the normal relationship between notes, key signatures
% and clefs:
\set Staff.clefGlyph = "clefs.F"
\set Staff.clefPosition = #2
\c'\break
\set Staff.clefGlyph = "clefs.G"
\c'\break
\set Staff.clefGlyph = "clefs.C"
\c'\break
\set Staff.clefTransposition = #7
\c'\break
\set Staff.clefTransposition = #0
\set Staff.clefPosition = #0
\c'\break
% Return to the normal clef:
\set Staff.middleCPosition = #0
\c'\break
}
Siehe auch

Notationsreferenz: [Mensurale Schlüssel], Seite 414, [Gregorianische Schlüssel], Seite 422, [Standardtabulaturen], Seite 322, [Angepasste Tabulaturen], Seite 335, [Stichnoten formatieren], Seite 198.

Schnipsel: Abschnitt “Pitches” in Schnipsel.


**Tonartbezeichnung**

_Achtung:_ Neue Benutzer sind manchmal verwirrt, wie Versetzungssymbole und Vorzeichen/Tonarten funktionieren. In LilyPond sind Notenbezeichnungen die wirkliche Tonhöhe, erst durch Vorzeichen wird bestimmt, wie diese Tonhöhe dann im Notenbild dargestellt wird. Eine einfache Tonhöhe wie etwa `c` bedeutet also immer das eingestrichene C ohne Versetzungssymbole, egal was für Vorzeichen/Tonart oder Schlüssel gesetzt sind. Mehr Information dazu in Abschnitt “Tonhöhen und Tonartbezeichnungen (Vorzeichen)” in Handbuch zum Lernen.

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.

Die Tonart kann geändert werden:

```lilypond
\key Tonhöhe Modus
```

Der Wert _Modus_ 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` (Lokrisch), `
eaolian` (Aeolisch), `
mixolydian` (Mixolydisch), `
lydian` (Lydisch), `
phrygian` (Phrygisch) und `
dorian` (Dorisch).

```lilypond
\relative { 
  \key g \major 
  fis'1 
  f 
  fis 
}
```

_Ausgewählte Schnipsel_

*Auflösungszeichen nicht setzen, wenn die Tonart wechselt*

Wenn die Tonart wechselt, werden automatisch Auflösungszeichen ausgegeben, um Versetzungssymbole der vorherigen Tonart aufzulösen. Das kann verhindert werden, indem die `printKeyCancellation`-Eigenschaft im Staff-Kontext auf "false" gesetzt wird.

```lilypond
\relative c' { 
  \key d \major 
  a4 b cis d 
  \key g \minor 
  a4 bes c d 
}
Kapitel 1: Musikalische Notation

\set Staff.printKeyCancellation = ##f
\key d \major
a4 b cis d
\key g \minor
a4 bes c d
}

Untypische Tonarten

Der üblicherweise benutzte \key-Befehl setzt die keySignature-Eigenschaft im Staff-Kontext.

Um untypische Tonartenvorzeichen zu erstellen, muss man diese Eigenschaft direkt setzen. Das Format für den Befehl ist eine Liste: \set Staff.keySignature = #`(((Oktave . Schritt) . Alteration) ((Oktave . Schritt) . Alteration) ...) wobei für jedes Element in der Liste Oktave die Oktave angibt (0 ist die Oktave vom eingestrichenen C bis zum eingestrichenen H), Schritt gibt die Note innerhalb der Oktave an (0 heißt C und 6 heißt H), und Alteration ist ,SHARP ,FLAT ,DOUBLE-SHARP usw. (Beachte das beginnende Komma.)

Alternativ kann auch jedes Element der Liste mit dem allgemeineren Format (Schritt . Alteration) gesetzt werden, wobei dann die Einstellungen für alle Oktaven gelten.

Hier ein Beispiel einer möglichen Tonart für eine Ganztonleiter:
\include "arabic.ly"
\relative do' {
\set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
(1 . ,SEMI-FLAT)
(2 . ,FLAT)
(5 . ,FLAT)
(6 . ,SEMI-FLAT))
%
\set Staff.extraNatural = ##f
re reb \dwn reb read
dod dob dosd \dwn dob |
dobsb dodsd do do |
}

Siehe auch

Glossar: Abschnitt “church mode” in Glossar, Abschnitt “scordatura” in Glossar.
Handbuch zum Lernen: Abschnitt “Tonhöhen und Tonartbezeichnungen (Vorzeichen)” in Handbuch zum Lernen.
Schnipsel: Abschnitt “Pitches” in Schnipsel.
Oktavierungsklammern

Oktavierungsklammern zeigen eine zusätzliche Transposition von einer Oktave an:

\relative a' {
    a2 b
    \ottava #-2
    a2 b
    \ottava #1
    a2 b
    \ottava #0
    a2 b
    \ottava #1
    a2 b
    \ottava #2
    a2 b
}

Auszgewählte Schnipsel

Ottava-Text

Intern setzt die set-octavation-Funktion die Eigenschaften ottavation (etwa auf den Wert "8va" oder "8vb") und middleCPosition. Um den Text der Oktavierungsklammer zu ändern, kann ottavation manuell gesetzt werden, nachdem set-octavation benützt wurde.

{
    \ottava #1
    \set Staff.ottavation = #"8"
    c''1
    \ottava #0
    c'1
    \ottava #1
    \set Staff.ottavation = #"Text"
    c''1
}

Siehe auch

Glossar: Abschnitt "octavation" in Glossar.
Schnipsel: Abschnitt "Pitches" in Schnipsel.
Referenz der Interna: Abschnitt "Ottava_spanner_engraver" in Referenz der Interna, Abschnitt "OttavaBracket" in Referenz der Interna, Abschnitt "ottava-bracket-interface" in Referenz der Interna.
Transposition von Instrumenten

Wenn man Noten setzt, die von transponierenden Instrumenten gespielt werden, sind oft einige Stimmen auf einer anderen Tonhöhe notiert als dem Kammerton. In diesem Fall muss die Tonart des transponierenden Instruments gekennzeichnet werden, weil sonst die MIDI-Ausgabe und Stichnoten in anderen Stimmen falsche Tonhöhen produzieren. Mehr Information zu Stichnoten in [Andere Stimmen zitieren], Seite 195.

\transposition Tonhöhe

Die Tonhöhe, die für \transposition benutzt wird, muss mit dem wirklichen Ton übereinstimmen, der erklingt, wenn das Instrument ein c' in seiner Stimme spielt. Die Tonhöhe wird im absoluten Modus angegeben, ein Instrument also, dass einen Ton höher erklingt als es notiert wird, muss folgenden Befehl benutzen: \transposition d'. \transposition sollte nur dann benutzt werden, wenn sie nicht nicht in C notiert werden.

Hier einige Noten für Geige und B-Klarinette: die Stimmen (Noten und Vorzeichen) sind so notiert, wie sie in der Partitur erscheinen. Die zwei Instrumente spielen unisono.

\new GrandStaff <<
\new Staff = "violin" {
\relative c'' {
\set Staff.instrumentName = "Vln"
\set Staff.midiInstrument = "violin"
% not strictly necessary, but a good reminder
\transposition c'
\key c \major
\g4( c8) r c r c4
}
}
\new Staff = "clarinet" {
\relative c'' {
\set Staff.instrumentName = \markup { Cl (B\flat) }
\set Staff.midiInstrument = "clarinet"
\transposition bes
\key d \major
\a4( d8) r d r d4
}
}
>>


\relative c' {
\set Staff.instrumentName = "Cl (A)"
\key a \major
Siehe auch

Glossar: Abschnitt “concert pitch” in Glossar, Abschnitt “transposing instrument” in Glossar.

Notationsreferenz: [Andere Stimmen zitieren], Seite 195, [Transponieren], Seite 10.
Schnipsel: Abschnitt “Pitches” in Schnipsel.

Automatische Versetzungszeichen

Es gibt viele unterschiedliche Regeln, wie Versetzungszeichen notiert werden. LilyPond hat eine Funktion, mit der spezifiziert werden kann, welcher Stil benutzt werden soll. Diese Funktion kann man wie folgt benutzen:

```lilypond
\new Staff <<
    \accidentalStyle voice
    { ... }
>>
```

Der Versetzungszeichenstil bezieht sich auf das aktuelle Notensystem in der Standardeinstellung (eine Ausnahme bilden die Stile piano und piano-cautionary, die weiter unten erklärt werden). Die Funktion kann aber auch ein zweites Argument erhalten, mit der spezifiziert wird, auf welchen Bereich sich der neue Stil erstreckt. Um etwa den neuen Stil in allen Systemen einer Stimmgruppe (StaffGroup) zu benutzen, müsste der Befehl so aussehen:

```lilypond
\accidentalStyle StaffGroup.voice
```

Folgende Versetzungszeichenstile sind unterstützt. Um jeden Stil zu erklären, wird folgendes Beispiel benützt:

```lilypond
musicA = {
    <<
        \relative { 
            cis'8 fis, bes'4 <a cis>8 f bis4 |
            cis2. <c, g'>4 |
        }
        \\
        \relative { 
            ais'2 cis, |
            fis8 b a4 cis2 |
        }
    >>
```
Die letzten Zeilen des Beispiels könnten auch mit folgendem Code ersetzt werden, solange der gleiche Versetzungszeichenstil in beiden Systemen benutzt werden soll:

```latex
\new PianoStaff {
  \context Staff = "up" {
    \accidentalStyle default
    \musicA
  }
  \context Staff = "down" {
    \accidentalStyle default
    \musicB
  }
}
```

Die Tastaturbeispiele können auch mit folgenden Versionen ausgetauscht werden, solange der gleiche Verzeichnungszustand in beiden Systemen benutzt wird:

```latex
\new PianoStaff {
  \context Staff = "up" {
    \accidentalStyle default
    \musicA
  }
  \context Staff = "down" {
    \accidentalStyle default
    \musicB
  }
```

Die Zeilen mit \texttt{\change Staff = up} und \texttt{\change Staff = down} können mit \texttt{\change Staff = up} und \texttt{\change Staff = down} ausgetauscht werden, solange der gleiche Verzeichnungszustand in beiden Systemen benutzt wird.

```latex
\new PianoStaff {
  \context Staff = "up" {
    \accidentalStyle default
    \musicA
  }
  \context Staff = "down" {
    \accidentalStyle default
    \musicB
  }
```
default (Standard)

Das ist das Standardverhalten. Es entspricht der Konvention für Notation von Musik des 18. Jahrhunderts: Versetzungszeichen werden bis zum Taktende erinnert, in dem sie gesetzt wurden, und nur in ihrer eigenen Oktave. Im nächsten Beispiel wird also kein Auslösender vor dem \( b \) (H) im zweiten Takt oder dem letzten \( c \) gesetzt:

![Notation Beispiel 1](image1)

voice (Stimme)

Das normale Verhalten ist es, die Versetzungszeichen auf der Notensystemebene zu erinnern. In diesem Stil aber werden Versetzungszeichen individuell für jede Stimme errechnet. Abgesehen davon gelten die Regeln des Standardstiles (default).

Das hat zur Folge, dass Versetzungszeichen von einer Stimme in der anderen nicht aufgelöst werden, was oft ein unerwünschtes Ergebnis ist: im folgenden Beispiel kann man schwer sagen, ob das zweite \( a \) unalterioriert oder erhöht gespielt werden soll. Die voice-Option sollte also nur benutzt werden, wenn die Stimmen separat von unterschiedlichen Musikern gelesen werden. Wenn das System nur von einem Musiker benutzt wird (etwa der Dirigent oder ein Klavierspieler), dann sind die Stile modern oder modern-cautionary besser.

![Notation Beispiel 2](image2)

modern (Modern)


![Notation Beispiel 3](image3)
Kapitel 1: Musikalische Notation

modern-cautionary (Modern mit Warnungen)
Dieser Stil ähnelt modern, aber die „zusätzlichen“ Versetzungszeichen (die normalerweise nicht gesetzt werden) werden als Warnungen gesetzt. In der Standardeinstellung werden sie in Klammern gesetzt, aber sie können auch in kleinerer Größe gesetzt werden, wenn man die cautionary-style-Eigenschaft von AccidentalSuggestion definiert.

modern-voice (Modern für Stimmen)
Diese Regel wird für viestimmige Noten benutzt, die sowohl von unterschiedlichen Spielern für jede Stimme als auch von einem Spieler für alle Stimmen benutzt. Versetzungszeichen werden für jede Stimme gesetzt, aber sie werden über die Stimme hinweg aufgelöst innerhalb des selben Notensystems. Das a im letzten Takt ist also aufgelöst, weil die vorigen Auflösung in einer anderen Stimme stattgefunden hatte, und das d im unteren System ist aufgelöst wegen eines Versetzungszeichens in einer anderen Stimme im vorigen Takt:

modern-voice-cautionary (modern mit Warnungen für einzelne Stimmen)
Dieser Stil ist der gleiche wie modern-voice, nur dass hier die zusätzlichen Versetzungszeichen (die nicht vom voice-Stil gesetzt werden) als Warnungsversetzungszeichen gesetzt werden. Obwohl alle Versetzungszeichen, die mit default gesetzt werden, auch mit diesem Stil gesetzt werden, sind manche Warnungsversetzungszeichen.
piano (Klavier)
Dieser Stil orientiert sich an den Regeln im 20. Jahrhundert für die Notation von Klavierspiel. Er ist sehr ähnlich mit dem modernen Stil, aber Versetzungspunkte werden auch über Notensysteme hinweg für die selbe Akkolade (GrandStaff oder PianoStaff) aufgelöst.
Dieser Versetzungspunktstil wirkt sich standardmäßig auf die gesamte Akkolade (GrandStaff oder PianoStaff) aus.

piano-cautionary (Klavier mit Warnungen)
Dieser Stil verhält sich wie piano, aber die zusätzlichen Versetzungspunkte werden als Warnungen ausgegeben:

neo-modern
Dieser Stil richtet sich nach den Regeln für moderne Musik: Versetzungspunkte werden mit im modern-Stil gesetzt, aber sie werden nochmal gesetzt, wenn die gleiche Note später im selben Takt auftritt – außer die Note wird unmittelbar wiederholt.

neo-modern-cautionary (neo-modern mit Warnungen)
Dieser Stil ähnelt neo-modern, aber die zusätzlichen Versetzungspunkte werden als Warnungen gesetzt.
neo-modern-voice (neo-modern für Stimmen)
Diese Regel wird für Versetzungszeichen in mehreren Stimmen eingesetzt, wenn die Noten sowohl von Musikern gelesen werden, die nur eine Stimme lesen, als auch von Musikern, die alle Stimmen lesen. Versetzungszeichen werden für jede Stimme so wie mit der neo-modern-Regel gesetzt, aber innerhalb des gesamten Notensystems mit Auflösungszeichen versehen.

neo-modern-voice-cautionary
Diese Regel ähnelt neo-modern-voice, aber die zusätzlichen Versetzungszeichen werden hier als warnende Versetzungszeichen gesetzt.

dodecaphonic (Zwölftonmusik)

teaching (didaktisch)
Dieser Stil ist für Lernende bestimmt: der Stil orientiert sich am modern-Stil, aber die Alterationen, die sich durch die Tonart ergeben, werden zusätzlich als Warnungsversetzungszeichen gesetzt. Eine Ausnahme sind direkt wiederholte Noten.
**no-reset (nicht zurücksetzen)**

Das ist der gleiche Stil wie default, aber die Versetzungszeichen dauern für „immer“ an, nicht nur im aktuellen Takt:

![Musiknotation Beispiel](image)

**forget (vergessen)**

Das ist das Gegenteil von no-reset: Versetzungszeichen werden überhaupt nicht erinnert und folgerichtig werden alle Versetzungszeichen entsprechend der Tonart gesetzt, unabhängig vom Kontext der Noten.

![Musiknotation Beispiel](image)

**Siehe auch**

Schnipsel: Abschnitt “Pitches” in Schnipsel.


**Bekannte Probleme und Warnungen**

Gleichzeitig erklingende Noten werden bei der automatischen Bestimmung der Versetzungszeichen nicht berücksichtigt: nur die vorige Note und die Vorzeichen werden einbezogen. Man muss die Versetzungszeichen mit ! oder ? schreiben, wenn gleichzeitig unterschiedliche Alterationen vorkommen, wie etwa für '&f! fis!'.

Die warnenden Auflösungszeichen werden gesetzt, indem die vorangegangenen Takte betrachtet werden. In der zweiten oder einer weiteren Wiederholungsklammer erwartet man jedoch, dass die Auflösungszeichen sich aus dem letzten gespielten und nicht dem letzten gesetzten Takt ergeben. Im folgenden Beispiel bräuchte das c in der zweiten Klammer kein Auflösungszeichen:

![Musiknotation Beispiel](image)

Die folgende Notlösung kann benutzt werden: Man definiert eine Funktion, die den Versetzungszeichenstil kurzzeitig auf forget umschaltet:

```scheme
forget = #(define-music-function (music) (ly:music?) #{
```
Tonumfang

Der Begriff *ambitus* (Pl. ambitus) beschreibt den Stimmumfang einer Stimme. Er kann auch
die Töne bedeuten, die ein Musikinstrument zu spielen in der Lage ist. Ambitus werden in
Chorpartituren gesetzt, damit die Sänger schnell wissen, ob sie die Stimme meistern können.

Ambitus werden zu Beginn des Stückes nahe des ersten Schlüssels notiert. Der Stimmum-
fang wird durch zwei Notenköpfe dargestellt, die die tiefste und höchste Note der Stimme
repräsentieren. Versetzungssignale werden nur gesetzt, wenn sie nicht durch die Tonart definiert
werden.

Ausgewählte Schnipsel

*Ambitus pro Stimme hinzufügen*

Ambitus können pro Stimme gesetzt werden. In diesem Fall müssen sie manuell verschoben
werden, um Zusammenstöße zu verhindern.

\new Staff <<
Kapitel 1: Musikalische Notation

Ambitus mit vielen Stimmen

Indem man den Ambitus_engraver im Staff-Kontext hinzufügt, erhält man einen einzigen Ambitus pro System, auch in dem Fall, dass mehrere Stimmen sich im gleichen System befinden.

Changing the ambitus gap

It is possible to change the default gap between the ambitus noteheads and the line joining them.

\consists "Ambitus_graver"
}
}
\new Staff {
 \time 2/4
 % Default setting
 c'4 g''
}
\new Staff {
 \time 2/4
 \override AmbitusLine.gap = #0
 c'4 g''
}
\new Staff {
 \time 2/4
 \override AmbitusLine.gap = #1
 c'4 g''
}
\new Staff {
 \time 2/4
 \override AmbitusLine.gap = #1.5
 c'4 g''
}

Siehe auch

Glossar: Abschnitt “ambitus” in Glossar.

Schnipsel: Abschnitt “Pitches” in Schnipsel.


Bekannte Probleme und Warnungen
Es gibt keine Kollisionskontrolle bei mehreren Ambitus in einem System.

1.1.4 Notenköpfe
Dieser Abschnitt zeigt, wie man Notenköpfe ändern kann.

Besondere Notenköpfe
Notenköpfe können verändert werden:
\relative { c'' { c4 b \override NoteHead.style = #'cross c4 b \revert NoteHead.style a b \override NoteHead.style = #'harmonic a b \revert NoteHead.style c4 d e f } }

Für alle Notenkopfstile siehe Abschnitt A.9 [Notenkopfstile], Seite 646.
Der Kreuz-(cross) Stil wird mit unterschiedlichen musikalischen Absichten eingesetzt. Die folgenden vordefinierten Befehle verändern die Notenköpfe sowohl in Notensystemen als auch in Tabulaturen und können benutzt werden, um alle musikalischen Bedeutungen zu notieren:
\relative { c''4 b \xNotesOn a b c4 b \xNotesOff c4 d }

Die Form als musikalische Funktion dieses Befehls kann innerhalb und außerhalb von Akkorden benutzt werden, um Notenköpfe mit Kreuzen in normalen und Tabulatursystemen zu erstellen:
\relative { c''4 b \xNote { e f } c b < g \xNote c f > b }
Als Synonym für \Note, \NotesOn und \NotesOff kann \deadNote, \deadNotesOn und \deadNotesOff benutzt werden. Der Begriff dean note (engl. tote Note) wird regelmäßig von Gitaristen benutzt.

Es gibt auch einen Kurzbefehl für die Rautenform, der nur innerhalb von Akkorden benutzt werden kann:

\begin{verbatim}
<\c f\harmonic>2 <\d a'\harmonic>4 <\c g'\harmonic>
\end{verbatim}

Vordefinierte Befehle
\harmonic, \NotesOn, \NotesOff, \Note.

Siehe auch
Schnipsel: Abschnitt “Pitches” in Schnipsel.


Easy-Notation-Notenköpfe
Die „einfachen Notenköpfe“ haben die Bezeichnung der Note im Kopf gedruckt. Das wird eingesetzt, um die Notation beizubringen. Damit die Buchstaben noch lesbar sind, müssen sie sehr groß gesetzt werden. Wie man eine größere Schriftart einstellt, findet sich in Abschnitt 4.2.2 [Die Notensystemgröße einstellen], Seite 511.

\begin{verbatim}
#(set-global-staff-size 26)
\relative c' {
  \easyHeadsOn
  c2 e4 f
  g1
  \easyHeadsOff
  c,1
}
\end{verbatim}

Vordefinierte Befehle
\easyHeadsOn, \easyHeadsOff.
Ausgewählte Schnipsel

Numbers as easy note heads

Easy notation note heads use the note-names property of the NoteHead object to determine what appears inside the note head. By overriding this property, it is possible to print numbers representing the scale-degree.

A simple engraver can be created to do this for every note head object it sees.

```scheme
#(define Ez_numbers_engraver
  (make-engraver
    (acknowledgers
      ((note-head-interface engraver grob source-engraver)
        (let* ((context (ly:translator-context engraver))
            (tonic-pitch (ly:context-property context 'tonic))
            (tonic-name (ly:pitch-notename tonic-pitch))
            (grob-pitch
              (ly:event-property (event-cause grob) 'pitch))
            (grob-name (ly:pitch-notename grob-pitch))
            (delta (modulo (- grob-name tonic-name) 7))
            (note-names
              (make-vector 7 (number->string (1+ delta)))))
          (ly:grob-set-property! grob 'note-names note-names))))
  (set-global-staff-size 26)

\layout {
  \ragged-right = ##t
  \context {
    \Voice
    \consists Ez_numbers_engraver
  }
}
\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break
  \key a \major
  a,4 b cis d
  e4 fis gis a \break
  \key d \dorian
  d,4 e f g
  a4 b c d
}
```

```
\times 4 \clef G
```

```
\accidental sharp
\accidental sharp
\accidental sharp
```

```
36
```

Kopf 1: Musikalische Notation
Notenköpfe mit besonderen Formen

In dieser Notation haben die Notenköpfe eine Form, die ihrer harmonischen Funktion innerhalb der Tonleiter entspricht. Die Notation war sehr beliebt in amerikanischen Liederbüchern des 19. Jahrhunderts. Auf diese Weise können die Formen Sacred Harp, Southern Harmony, Funk (Harmonia Sacra), Walker und Aiken (Christian Harmony) benutzt werden:

```
\relative c' {
  \aikenHeads
  c, d e f g2 a b1 c \break
  \sacredHarpHeads
  c,4 d e f g2 a b1 c \break
  \southernHarmonyHeads
  c,4 d e f g2 a b1 c \break
  \funkHeads
  c,4 d e f g2 a b1 c \break
  \walkerHeads
  c,4 d e f g2 a b1 c \break
}
```

Siehe auch

Notationsreferenz: Abschnitt 4.2.2 [Die Notensystemgröße einstellen], Seite 511.
Schnipsel: Abschnitt “Pitch” in Schnipsel.

Die unterschiedlichen Formen richten sich nach der Stufe in der Skala, wobei der Grundton der Skala aus dem \key-Befehl entnommen wird. Wenn man eine Moll-Skala benutzt, ergibt sich die Form der Notenköpfe aus der parallelen Dur-Tonleiter:
\begin{verbatim}
\key a \minor
\aikenHeads
a b c d e2 f g1 a \break
\aikenHeadsMinor
a,4 b c d e2 f g1 a \break
\sacredHarpHeadsMinor
a,2 b c d \break
\southernHarmonyHeadsMinor
a2 b c d \break
\funkHeadsMinor
a2 b c d \break
\walkerHeadsMinor
a2 b c d \break
\end{verbatim}

Vordefinierte Befehle
\aikenHeads, \aikenHeadsMinor, \funkHeads, \funkHeadsMinor, \sacredHarpHeads, \sacredHarpHeadsMinor, \southernHarmonyHeads, \southernHarmonyHeadsMinor, \walkerHeads, \walkerHeadsMinor.
Ausgewählte Schnipsel

Notenkopfstile basierend auf der Tonleiterstufe erstellen

Die `shapeNoteStyles`-(NotenFormenStile)-Eigenschaft kann benutzt werden, um verschiedene Notenstile für jeden Schritt der Tonleiter zu definieren (vorgegeben von der Tonart oder der „tonic“ (Tonika)-Eigenschaft. Diese Eigenschaft braucht eine Anzahl von Symbolen, welche beliebig sein können (geometrische Ausdrücke wie `triangle` (Dreieck), `cross` (Kreuz) und `xcircle` (X-Kreis) sind erlaubt) oder basierend auf einer alten amerikanischen Notensatztradition (einige lateinische Notenbezeichnungen sind auch erlaubt).

Um alte amerikanische Liederbücher zu imitieren, gibt es einige vordefinierte Notenstile wie etwa `\aikenHeads` (im Stil von Aiken) oder `\sacredHarpHeads` (im Stil der Sacred Harp-Tradition).

Dieses Beispiel zeigt, wie man unterschiedlich geformte Noten erhält und eine Melodie transponieren kann, ohne dass das Verhältnis zwischen den harmonischen Funktionen und dem Notenstil verloren geht.

```latex
\begin{Verbatim}
fragment = {
  \key c \major
  c2 d
e2 f
g2 a
b2 c
}
\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
       #f la ti)
    \fragment
  }
  \break
  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
       mensural xcircle diamond)
    \fragment
  }
}
\end{Verbatim}
```

Alle Notenkopfstile finden sich in Abschnitt A.9 [Notenkopfstile], Seite 646.
Siehe auch

Schnipsel: Abschnitt “Pitches” in Schnipsel.

Notationsreferenz: Abschnitt A.9 [Notenkopfstile], Seite 646.


Improvisation

Improvisation wird manchmal angezeigt, indem schräge Notenköpfe gesetzt werden, wenn der Spieler eine beliebige Tonhöhe wählen kann aber den vorgegebenen Rhythmus spielen soll. Sie können wie folgt benutzt werden:

\new Voice \with {
  \consists "Pitch_squash_engraver"
} \relative {
  e''8 e g a a16( bes) a8 g
  \improvisationOn
  e8 ~
  2 ~ 8 f4 f8 ~
  2
  \improvisationOff
  a16( bes) a8 g e
}

\improvisationOn, \improvisationOff.

Vordefinierte Befehle

\improvisationOn, \improvisationOff.

Siehe auch

Schnipsel: Abschnitt “Pitches” in Schnipsel.


1.2 Rhythmus
Dieser Abschnitt erklärt die Eingabe von Rhythmen, Pausen, Dauern, Bebalkung und Takten.

1.2.1 Rhythmen eingeben

Tondauern

Notenlängen (Dauern) werden durch Zahlen und Punkte notiert: Dauern werden als reziproke Werte geschrieben. Zum Beispiel wird eine Viertelnote mit $\frac{4}{2}$ notiert (weil sie eine 1/4-Note ist), eine halbe Note mit $\frac{2}{1}$ (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. Notendauern bis hin zu 128steln sind unterstützt. Kürzere Notenwerte können auch notiert werden, können allerdings nur als Noten mit Balken auftreten.

\begin{verbatim}
\relative {
  \time 8/1
  \autoBeamOff
  \breve c' \longa c \breve c1 c2
  c4 c8 c16 c32 c64 c128 c128
}
\end{verbatim}

Hier die selben Notendauern ohne die Balken.

\begin{verbatim}
\relative {
  \time 8/1
  \longa c' \longa c \breve c1 c2
  c4 c8 c16 c32 c64 c128 c128
}
\end{verbatim}
Eine Note mit der vierfachen Dauer einer Brevis kann mit dem Befehl \maxima eingegeben werden, aber ihre Darstellung ist nur für die Alte Musiknotation unterstützt. Zu Einzelheiten siehe Abschnitt 2.9 [Notation von alter Musik], Seite 409.

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.

\relative { 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, usw.

\relative { a'4 b c4. b8 a4. b4.. c8. }

Manche Notenlängen können nicht mit binären Dauern und Punkten dargestellt werden, sie können nur erreicht werden, indem man Noten überbindet. Für Einzelheiten siehe [Bindebögen], Seite 49.


Optional können Noten streng proportional nach ihrer exakten Dauer gesetzt werden. Zu Einzelheiten hierzu und weiteren Einstellungen für proportionale Notation siehe Abschnitt 4.5.5 [Proportionale Notation], Seite 541.

Punkte werden normalerweise nach oben verschoben, damit sie die Notenlinien nicht berühren. Punkte können manuell über oder unter dem Notensystem gesetzt werden, zu Einzelheiten siehe Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

Vordefinierte Befehle
\autoBeamOn, \autoBeamOff, \dotsUp, \dotsDown, \dotsNeutral.

Ausgewählte Schnipsel

Alternative breve notes
Breve notes are also available with two vertical lines on each side of the notehead instead of one line and in baroque style.

\relative c' {'
    \time 4/2
    c\breve |
    \override Staff.NoteHead.style = #'altdefault
    b\breve
}
Changing the number of augmentation dots per note
The number of augmentation dots on a single note can be changed independently of the dots placed after the note.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = #4
  c4.. a16 r2 |
  \override Dots.dot-count = #0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}
```

Siehe auch

Notationsreferenz: [Automatische Balken], Seite 77, [Bindebogen], Seite 49, [Hälsen], Seite 210, Abschnitt 1.2.1 [Rhythmen eingeben], Seite 41, Abschnitt 1.2.2 [Pausen eingeben], Seite 52, Abschnitt 2.1 [Notation von Gesang], Seite 241, Abschnitt 2.9 [Notation von alter Musik], Seite 409, Abschnitt 4.5.5 [Proportionale Notation], Seite 541.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.


Bekannte Probleme und Warnungen
Es gibt keine grundlegende Grenze für die Dauer von Pausen (sowohl kürzer als auch länger), aber die Anzahl an Symbolen ist begrenzt: Einzelne Pausen können von 128stel bis zur Maxima (8 Ganze) gesetzt werden.

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 (oder unter) den Noten ausgegeben, optional mit einer eckigen Klammer, die
Kapitel 1: Musikalische Notation

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.

\begin{verbatim}
a2 \tuplet 3/2 { b4 b b } 
c4 c \tuplet 3/2 { b4 a g }
\end{verbatim}

Triolenklammern können manuell über oder unter dem Notensystem ausgegeben werden, siehe Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

N-tolen können ineinander geschachtelt werden:

\begin{verbatim}
\relative {
    \autoBeamOff 
c''4 \tuplet 5/4 { f8 e f \tuplet 3/2 { e[ f g] } } f4 }
\end{verbatim}

Wenn man die Eigenschaften von N-tolen verändern will, die zum selben musikalischen Zeitpunkt beginnen, muss \tweak eingesetzt werden.

Um die Dauern von Noten zu ändern, ohne die N-tolen-Klammern zu setzen, siehe [Tondauern skalieren], Seite 48.

Verdefinierte Befehle

\tupletUp, \tupletDown, \tupletNeutral.

Ausgewählte Schnipsel

\textit{Entering several tuplets using only one \tuplet command}

The property \tupletSpannerDuration sets how long each of the tuplets contained within the brackets after \tuplet should last. Many consecutive tuplets can then be placed within a single \tuplet expression, thus saving typing.

There are several ways to set \tupletSpannerDuration. The command \tupletSpan sets it to a given duration, and clears it when instead of a duration \default is specified. Another way is to use an optional argument with \tuplet.

\begin{verbatim}
\relative c' {
    \time 2/4 
    \tupletSpan 4 
    \tuplet 3/2 { c8"\tupletSpan 4" c c c c c }
    \tupletSpan \default 
    \tuplet 3/2 { c8"\tupletSpan \default" c c c c c }
    \tuplet 3/2 4 { c8"\tupletSpan 3/2 4 {...}" c c c c c }
}
\tuplet 3/2 4 {...}
\end{verbatim}
Die Zahl der N-töne verändern

Standardmäßig wird nur der Zähler des N-tolen-Bruchs über der Klammer dargestellt, wie er dem \times-Befehl übergeben wird. Man kann aber auch Zähler/Nenner ausgeben lassen, oder die Zahl vollständig unterdrücken.

\relative c' {
\tuplet 3/2 { c8 c c }
\tuplet 3/2 { c8 c c }
\override TupletNumber.text = #tuplet-number::calc-fraction-text
\tuplet 3/2 { c8 c c }
\omit TupletNumber
\tuplet 3/2 { c8 c c }
}

\begin{musicfigure}
\begin{musicstaves}
\begin{musicstaves}
\begin{musicstaff}
\relative c' {
\once \override TupletNumber.text = #(tuplet-number::non-default-tuplet-denominator-text 7)
\tuplet 3/2 { c4. c4. c4. c4. }
\once \override TupletNumber.text = #(tuplet-number::non-default-tuplet-fraction-text 12 7)
\tuplet 3/2 { c4. c4. c4. c4. }
\once \override TupletNumber.text = #(tuplet-number::append-note-wrapper
\tuplet-number::calc-denominator-text "4")
\tuplet 3/2 { c8 c8 c8 c8 c8 c8 }
\once \override TupletNumber.text = #(tuplet-number::append-note-wrapper
\tuplet-number::calc-fraction-text "4")
\tuplet 3/2 { c8 c8 c8 c8 c8 }
\once \override TupletNumber.text = #(tuplet-number::fraction-with-notes "4." "8")
\tuplet 3/2 { c4. c4. c4. c4. }
\once \override TupletNumber.text = #(tuplet-number::non-default-fraction-with-notes 12 "8" 4 "4")
\tuplet 3/2 { c4. c4. c4. c4. }
}
\end{musicstaff}
\begin{musicstaff}
\begin{musicstaves}
\begin{musicstaves}
\begin{musicstaves}
\begin{musicstaff}
\overridetext
\end{musicstaff}
\end{musicstaves}
\end{musicstaves}
\end{musicstaves}
\end{musicstaff}
\end{musicstaves}
\end{musicstaves}
\end{musicfigures}
Controlling tuplet bracket visibility

The default behavior of tuplet-bracket visibility is to print a bracket unless there is a beam of the same length as the tuplet. To control the visibility of tuplet brackets, set the property 'bracket-visibility' to either #t (always print a bracket), #f (never print a bracket) or #’if-no-beam (only print a bracket if there is no beam).

```plaintext
music = \relative c'' { 
  \tuplet 3/2 { c16[ d e } f8] 
  \tuplet 3/2 { c8 d e } 
  \tuplet 3/2 { c4 d e }
}

\new Voice { 
  \relative c' { 
    << \music s4"default" >> 
    \override TupletBracket.bracket-visibility = #'if-no-beam 
    << \music s4"if-no-beam" >> 
    \override TupletBracket.bracket-visibility = ##t 
    << \music s4"#t" >> 
    \override TupletBracket.bracket-visibility = ##f 
    << \music s4"#f" >> 
    \%\% v2.18: 
    \omit TupletBracket 
    << \music s4"omit" >> 
  }
}
```
Zeilenumbrüche bei N-tolen mit Balken erlauben


\layout {
    \context {
        \Voice
        \remove "Forbid_line_break_engraver"
        \override Beam.breakable = ##t
    }
}
\relative c'' {
    \repeat unfold 5 { \tuplet 3/2 { c[ b a] } }
    \tuplet 3/2 { c[ b \bar "" \break a] }
    \repeat unfold 5 { \tuplet 3/2 { c[ b a] } }
    c8
}

Siehe auch


Notation referenz: [Verwaltung der Zeiteinheiten], Seite 111, [Tondauern skalieren], Seite 48, Abschnitt 5.3.4 [Der tweak-Befehl], Seite 576, [Polymetrische Notation], Seite 71.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.


Bekannte Probleme und Warnungen

Verzierungen können innerhalb von Triolenklammern gesetzt werden, außer wenn ein System mit einer Verzierung beginnt, die von einer N-tole gefolgt wird. In diesem besonderen Fall müssen die Verzierungen vor dem \times-Befehl gesetzt werden, damit sich keine Fehler ergeben.

**Tondauern skalieren**

Die Dauer von einzelnen Noten, Pausen oder Akkorden kann mit einem Bruch multipliziert werden, indem hinter die Notendauer "$N/M" (oder "$N" wenn $M$ 1 ist) geschrieben wird. Die Erscheinung der Noten oder Pausen wird dadurch nicht beeinflusst, die neue Dauer wird aber dazu benutzt, ihre Position im Takt zu errechnen und die neue Dauer in der MIDI-Ausgabe einzusetzen. Die Faktoren, mit denen multipliziert wird, können auch kombiniert werden, etwa "$L*M*/N". Die Faktoren sind Teil der Dauer: wenn man keine Dauer für die nächste Note angibt, wird die Dauer der vorigen Note mit allen Skalierungsfaktoren übernommen.

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

\begin{verbatim}
\relative { 
  \time 2/4 
  % Alter durations to triplets 
  a'4*2/3 gis a 
  % Normal durations 
  a4 a 
  % Double the duration of chord 
  <a d>4*2 
  % Duration of quarter, appears like sixteenth 
  b16*4 c4 
}
\end{verbatim}

Die Dauer von unsichtbaren Pausen kann auch mit einem Faktor beeinflusst werden. Das ist sinnvoll, wenn man viele Takte überspringen muss, etwa $s1*23$.

Längere Notenabschnitte können auf die gleiche Art durch Multiplikation mit einem Bruch komprimiert werden, als ob jede Note, jeder Akkord oder jede Pause mit dem Bruch multipliziert würde. Damit bleibt das Aussehen der Musik unverändert, aber die interne Dauer der Noten wird mit dem Bruch multipliziert. Hier ein Beispiel, das zeigt, wie Noten komprimiert und ausgedehnt werden kann:

\begin{verbatim}
\time 2/4 
% Normal durations 
<c a>4 c8 a 
% Scale music by *2/3 
\scaleDurations 2/3 { 
  <c a f>4. c8 a f 
} 
% Scale music by *2 
\scaleDurations 1/2 { 
  <c' a>4 c8 b 
}
\end{verbatim}

Eine Anwendung für diesen Befehl ist polymetrische Notation, siehe [Polymetrische Notation], Seite 71.
Siehe auch

Notationsreferenz: [Andere rhythmische Aufteilungen], Seite 43, [Unsichtbare Pausen], Seite 54, [Polymetrische Notation], Seite 71.
Schnipsel: Abschnitt “Rhythms” in Schnipsel.

Bindebögen

Ein Bindebogen verbindet zwei benachbarte Noten der selben Tonhöhe. Als Resultat wird die Dauer der Notenlänge verlängert.

Achtung: Bindebögen (engl. tie) dürfen nicht mit Legatobögen (engl. slur) verwechselt werden, durch die die Vortragsart bezeichnet wird, noch mit Phrasierungsbögen (engl. phrasing slur), die musikalische Phrasen anzeigen. Ein Bindebogen ist nur eine Art, die Tondauer zu verlängern, ähnlich etwa wie die Punktierung.

Ein Bindebogen wird mit der Tilde ~ (AltGr++) notiert.

\[ a2 \sim 2 \]

Bindebögen werden eingesetzt, wenn die Note entweder über eine Taktlinie hinüberreicht, oder wenn die entsprechende Dauer der Note nicht mit Punktierung erreicht werden kann. Bindebögen sollten auch benutzt werden, wenn Notenwerte über die inneren Unterteilungen von Takten hinüberreichen:

Wenn viele Noten über Taktlinien gebunden werden müssen, kann es einfacher sein, automatische Notenaufteilung einzustellen, wie beschrieben in [Automatische Aufteilung von Noten], Seite 74. Mit diesem Mechanismus werden lange Noten automatisch aufgeteilt, wenn sie über Taktgrenzen reichen.


\[ \langle c\ e\ g\rangle \sim \langle c\ e\ g\rangle \]
\[ \langle c^-e\ g^-b\rangle \sim \langle c\ e\ g\ b\rangle \]

Wenn die zweite Variante einer Wiederholung mit einer übergebundenen Note anfängt, muss der Bindebogen wie folgt notiert werden:

\[
\texttt{\textbackslash repeat volta 2 \{ c\ g\ \langle c\ e\rangle 2 \sim \}}
\texttt{\textbackslash alternative \{}
\]
% First alternative: following note is tied normally
{ <c e>2, r4 }

% Second alternative: following note has a repeated tie
{ <c e>2\repeatTie d4 c } }

\dots\dot
\dots\dot
\noteheads.s1
\noteheads.s1
\noteheads.s1/noteheads.s1
\clefs.G
\timesig.C44

1.
\noteheads.s2
\noteheads.s2
\noteheads.s2
\noteheads.s2
\timesig.C44
\clefs.G

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 folgendermaßen notiert werden:

\<c' f' g'>1\laissezVibrer

\dots\dot
\dots\dot
\noteheads.s1
\noteheads.s1
\noteheads.s1
\noteheads.s1
\clefs.G
\timesig.C44

Bindebögen können manuell über oder unter dem Notensystem gesetzt werden. Zu Einzelheiten siehe Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

Bindebögen können durchgehend, gestrichelt, gepunktet oder in einer Kombination von Strichen und durchgehender Linie definiert werden.

\tieDotted
c2 ~ 2
\tieDashed
c2 ~ 2
\tieHalfDashed
c2 ~ 2
\tieHalfSolid
c2 ~ 2
\tieSolid
c2 ~ 2

\tieDashPattern #0.3 #0.75
c2 ~ 2
\tieDashPattern #0.7 #1.5
c2 ~ 2
\tieSolid
c2 ~ 2

Eigene Strichelungsmuster können definiert werden:

\tieDashPattern #0.3 #0.75
c2 ~ 2
\tieDashPattern #0.7 #1.5
c2 ~ 2
\tieSolid
c2 ~ 2
Die Definition von Muster für die Strichelung der Bindebögen hat die gleiche Struktur wie die Definition für Legatobögen. Zu weiterer Information zu komplizierten Strichelungsmustern, siehe die Schnipsel im Abschnitt [Legatobögen], Seite 123.

Durch Veränderung der Eigenschaften `whiteout` (weiß malen) und `layer` (Ebene) kann verhindert werden, dass Bindebögen mit anderen Objekten im Notensystem zusammenstoßen.

\relative {
\override Tie.layer = #-2
\override Staff.TimeSignature.layer = #-1
\override Staff.KeySignature.layer = #-1
\override Staff.TimeSignature.whiteout = ##t
\override Staff.KeySignature.whiteout = ##t
b'2 b~
\time 3/4
\key a \major
b r4
}

Vordefinierte Befehle

`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieDashPattern`, `\tieHalfDashed`, `\tieHalfSolid`, `\tieSolid`.

Ausgewählte Schnipsel

Überbindungen für Arpeggio benutzen

Überbindungen werden teilweise benutzt, um Arpeggios zu notieren. In diesem Fall stehen übergebundenen Noten nicht unbedingt hintereinander. Das Verhalten kann erreicht werden, indem die `tieWaitForNote`-Eigenschaft auf `#t` gesetzt wird. Diese Funktion ist auch sinnvoll, um etwa ein Tremolo mit einem Akkord zu überbinden, kann aber prinzipiell auch für normale Überbindungen eingesetzt werden.

\relative c' {
\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
\tieUp
c8 ~ a
\tieDown
\tieDotted
g8 ~ c g2
}

\relative c' {
\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
\tieUp
c8 ~ a
\tieDown
\tieDotted
g8 ~ c g2
}
Bindebögen manuell setzen

Überbindungen können manuell gesetzt werden, indem man die tie-configuration-Eigenschaft des TieColumn-Objekts beeinflusst. Die erste Zahl zeigt den Abstand von der Mitte in Notensystemabständen an, die zweite Zahl zeigt die Richtung an (1 = nach oben, -1 = nach unten).

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

Siehe auch

Glossar: Abschnitt “tie” in Glossar, Abschnitt “laissez vibrer” in Glossar.
Notationsreferenz: [Legatobögen], Seite 123, [Automatische Aufteilung von Noten], Seite 74.

Bekannte Probleme und Warnungen
Der Wechsel zwischen Systemen bei aktiver Überbindung produziert keinen gekrümmten Bogen.

1.2.2 Pausen eingeben
Pausen werden als Teil der musikalischen Ausdrücke zusammen mit den Noten notiert.

Pausen
Pausen werden wie Noten eingegeben, ihre Bezeichnung ist r. Dauern, die länger als eine Ganze sind, haben die vordefinierten Befehle:

\new Staff {
    % These two lines are just to prettify this example
    \time 16/1
    \omit Staff.TimeSignature
    % Print a maxima rest, equal to four breves
    r\maxima
    % Print a longa rest, equal to two breves
    r\longa
    % Print a breve rest
    r\breve
    r1 r2 r4 r8 r16 r32 r64 r128
}
Pausen, die ganze Takte ausfüllen und in der Takmitte 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 [Ganztaktpausen], Seite 56.

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

\relative { a'4\rest d4\rest }

Ausgewählte Schnipsel

Pausenstile

Pausen können in verschiedenen Stilen dargestellt werden.

\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

  \override Staff.Rest.style = #'mensural
  r\maxima\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'neomensural
  r\maxima\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'classical
  r\maxima\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'z
  r\maxima\markup \typewriter { z-style }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'default
  r\maxima\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
}
Siehe auch


Notationsreferenz: [Ganztaktpausen], Seite 56.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.

Referenz der Interna: Abschnitt “Rest” in Referenz der Interna.

Bekannte Probleme und Warnungen

Es gibt keine grundlegende Grenze für die Dauer von Pausen (sowohl kürzer als auch länger), aber die Anzahl von Symbolen ist begrenzt: Es gibt Zeichen für Pausen von einer 128 bis zu einer Maxima (8 Ganze).

Unsichtbare Pausen

Eine unsichtbare Pause (auch als „skip“ oder Übersprung bezeichnet) kann wie eine Note eingegeben werden, die Notationsbezeichnung ist s.

\[ a4 \ a4 \ a4 \ s4 \ a4 \ \textbackslash{skip} \ 1 \ a4 \]

Die s-Syntax steht nur im Noten- oder Akkordmodus zur Verfügung. In anderen Situationen, z. B. innerhalb eines Liedtextes, muss der Befehl \textbackslash{skip} benutzt werden. \textbackslash{skip} benötigt eine explizite Dauerangabe.

\[
\{ \\
\quad a'2 \ \textbackslash{skip}2 \ a'2 \ a'2 \\
\} \\
\textbackslash{new \ Lyrics \ {}} \\
\quad \textbackslash{lyricmode \ {}} \\
\quad \ foo2 \ \textbackslash{skip} \ 1 \ bla2
\]

\[
\{ \\
\quad a'2 \ \textbackslash{skip}2 \ a'2 \ a'2 \\
\} \\
\textbackslash{new \ Lyrics \ {}} \\
\quad \textbackslash{lyricmode \ {}} \\
\quad \ foo2 \ \textbackslash{skip} \ 1 \ bla2
\]
Weil \texttt{\textbackslash skip} ein Befehl ist, wirkt er sich nicht auf die Dauer der folgenden Noten aus, anders als \texttt{s}.

\begin{verbatim}
<<
{ \repeat unfold 8 \{ a'4 \}
{ a'4 \textbackslash skip 2 a' |
  s2 a'
}
>>
\end{verbatim}

Die Platzhalterpause mit \texttt{s} erstellt \texttt{Staff-} und \texttt{Voice-}Kontext, wenn es erforderlich ist, genau so wie Noten und Pausen.

\begin{verbatim}
\{ s1 s s \}
\end{verbatim}

Der Überspringungsbefehl (\texttt{\textbackslash skip}) ist einfach ein leerer Platzhalter. Durch ihn wird überhaupt nichts gesetzt, auch keine transparenten Objekte.

\begin{verbatim}
\% This is valid input, but does nothing
\{ \textbackslash skip 1 \textbackslash skip 1 \textbackslash skip 1 \}
\end{verbatim}

\textbf{Siehe auch}

Handbuch zum lernen: Abschnitt “Sichtbarkeit und Farbe von Objekten” in \textit{Handbuch zum Lernen}.

Notationsreferenz: [Unsichtbare Noten], Seite 207, Abschnitt 5.4.6 [Sichtbarkeit von Objekten], Seite 589.

Schnipsel: Abschnitt “Rhythms” in \textit{Schnipsel}.

Referenz der Interna: Abschnitt “SkipMusic” in \textit{Referenz der Interna}
**Ganztaktpausen**

Pausen für einen oder mehrere ganze Takte werden wie Noten eingegeben, wobei die Bezeichnung ein Großbuchstabe **R** ist:

% Rest measures contracted to single measure
\compressFullBarRests
R1\cdot4
R1\cdot24
R1\cdot4
b2"Tutti" b4 a4


\compressFullBarRests
\time 2/4
R1 | R2 |
\time 3/4
R2. | R2.*2 |
\time 13/8
R1\cdot13/8 | R1\cdot13/8\cdot12 |
\time 10/8
R4\cdot5\cdot4 |

Eine Ganztaktpause wird abhängig von der Taktart entweder als Ganze oder Brevis-Pause gesetzt, zentriert im Takt.

\time 4/4
R1 |
\time 6/4
R1\cdot3/2 |
\time 8/4
R1\cdot2 |

In den Standardeinstellungen werden mehrtaktige Pausen ausgeschrieben gesetzt, sodass sie die entsprechende Anzahl von Takten einnehmen. Alternativ kann die mehrtaktige Pause aber
Kapitel 1: Musikalische Notation

...
Wenn eine Mehrtaktpause direkt auf einen Auftakt mit \texttt{\partial} folgt, werden möglicherweise daraus resultierende Taktprüfungswarnungen nicht angezeigt.

**Vordefinierte Befehle**
\texttt{\textLengthOn}, \texttt{\textLengthOff}, \texttt{\fermataMarkup}, \texttt{\compressFullBarRests}, \texttt{\expandFullBarRests}.

**Ausgewählte Schnipsel**

*Die Erscheinung von Pausentakten ändern*

Wenn zehn oder weniger Pausentakte vorkommen, wird eine Reihe von Longa- und Brevis- pausen (auch Kirchenpausen genannt) gesetzt, bei mehr Takten wird eine Linie mit der Tak- tanzahl ausgegeben. Der vorgegebene Wert von zehn kann geändert werden, indem man die \texttt{expand-limit-Eigenschaft} setzt:

\begin{verbatim}
\relative c'' {
\compressMMRests {
 R1*2 | R1*5 | R1*9
 \override MultiMeasureRest.expand-limit = #3
 R1*2 | R1*5 | R1*9
 }
}
\end{verbatim}

Positionierung von Ganztaktpausen

Anders als bei normalen Pausen gibt es keinen direkten Befehl, um die vertikale Position von Ganztaktpausen zu beeinflussen, indem man sie an eine Tonhöhe anhängt. In polyphoner Notation wird aber dennoch die Position der Pausen von geraden und ungeraden Stimmen voneinander unterschieden. Die Position von Ganztaktpausen kann wie folgt verändert werden:

\begin{verbatim}
\relative c'' {
 % Multi-measure rests by default are set under the fourth line
 R1
 % They can be moved using an override
 \override MultiMeasureRest.staff-position = #-2
 R1
 \override MultiMeasureRest.staff-position = #0
 R1
 \override MultiMeasureRest.staff-position = #2
 R1
 \override MultiMeasureRest.staff-position = #3
 R1
 \override MultiMeasureRest.staff-position = #6
 R1
 \revert MultiMeasureRest.staff-position
 \break
\end{verbatim}
% In two Voices, odd-numbered voices are under the top line
<< { R1 } \ { a1 } >>
% Even-numbered voices are under the bottom line
<< { a1 } \ { R1 } >>
% Multi-measure rests in both voices remain separate
<< { R1 } \ { R1 } >>

% Separating multi-measure rests in more than two voices
% requires an override
<< { R1 } \ \ { R1 } \ \
\once \override MultiMeasureRest.staff-position = #0
{ R1 }
>>

% Using compressed bars in multiple voices requires another override
% in all voices to avoid multiple instances being printed
\compressMMRests
<<
\revert MultiMeasureRest.direction
{ R1*3 }
\-
\revert MultiMeasureRest.direction
{ R1*3 }
>>

Textbeschriftung und Mehrtaktpausen

Textbeschreibungen, die an Mehrtaktpausen gehängt wird, wird über oder unter der Pause zentriert. Lange Beschreibungen lassen den Takt nicht breiter werden. Um eine Mehrtaktpause einer Beschriftung anzupassen, muss eine unsichtbare Pause mit der Beschriftung direkt vor der Mehrtaktpause eingesetzt werden.

Man sollte beachten, dass unsichtbare Pausen automatische Taktstriche nach sich ziehen. Text, der an eine unsichtbare Pause gehängt wird, ist links ausgerichtet an der Position, wo die Pause erscheinen würde. Wenn aber die Länge des Taktes durch die Länge des Textes bestimmt wird, sieht es so aus, als ob der Text zentriert gesetzt ist.

\relative c' {
\compressMMRests {
\textLengthOn
<>^\markup { [MAJOR GENERAL] }
R1*19
<>_\markup { \italic { Cue: ... it is yours } }
<>^\markup { A }
}
Siehe auch
Glossar: Abschnitt “multi-measure rest” in Glossar.
Notationsreferenz: [Tondauern], Seite 41, Abschnitt 1.8 [Text], Seite 215, Abschnitt 1.8.2 [Text formatieren], Seite 223, [Textarten], Seite 216.
Schnipsel: Abschnitt “Rhythms” in Schnipsel.
Referenz der Interna: Abschnitt “MultiMeasureRest” in Referenz der Interna, Abschnitt “MultiMeasureRestNumber” in Referenz der Interna, Abschnitt “MultiMeasureRestText” in Referenz der Interna.

Bekannte Probleme und Warnungen
Wenn man versucht, mit Fingersatz (etwa R1\*10-4) Zahlen über Ganztaktpausen zu setzen, kann die Zahl des Fingersatzes (4) mit der Taktanzahl (10) zusammenstoßen.
Es gibt keine Möglichkeit, normale Pausen automatisch zu Ganztaktpausen zu reduzieren.
Ganztaktpausen werden bei der Vermeidung von Zusammenstößen nicht berücksichtigt.

1.2.3 Rhythmen anzeigen lassen
Taktangabe
Taktangaben werden wie folgt erstellt.
\time 2/4 c' 2
\time 3/4 c' 2.

Taktangaben werden zu Beginn eines Stücks gesetzt und immer dann, wenn sich die Taktart ändert. Wenn eine Änderung am Ende einer Zeile geschieht, wird eine warnende Taktangabe am Ende der Zeile ausgegeben. Dieses Verhalten kann verändert werden, siehe Abschnitt 5.4.6 [Sichtbarkeit von Objekten], Seite 589.
Das Symbol für die Taktarten 2/2 und 4/4 kann in ein Zahlensymbol umgewandelt werden:

\relative c' {
  \time 2/2 c1
  \time 4/4 c1
  \numericTimeSignature
  \time 2/2 c1
  \time 4/4 c1
  \defaultTimeSignature
  \time 2/2 c1
}

Symbole für Modus und Proprietas der mensuralen Notation werden behandelt unter [Mensurale Taktartenbezeichnungen], Seite 415.


\score {
  \new Staff {
    \overrideTimeSignatureSettings
    \time 4/4
    \Override timeSignatureFraction = 4/4
    \Override baseMomentFraction = 1/4
    \Override beatStructure = 3,1
    \Override beamExceptions = #'
    \time 4/4
  }
}
\overrideTimeSignatureSettings braucht fünf Argumente:

1. \timeSignatureFraction (Taktart-Bruch), ein Bruch, der den Takt beschreibt.
2. \baseMomentFraction (Grundmoment-Bruch), ein Bruch, der den Zähler und Nenner der Grundschlageinheit der Taktart enthält.
3. \beatStructure (Taktzeit-Struktur), eine Scheme-Liste, die die Struktur der Taktschläge anzeigt, in Einheiten des Grundmoments.
4. \beamExceptions (Balken-Ausnahmen), eine Aliste, die alle Bebalkungsregeln für die Taktart enthält, außer dem Balken, der zum Taktende endet, wie beschrieben in [Einstellung von automatischen Balken], Seite 80.

Der Kontext, der \overrideTimeSignatureSettings enthält, muss begonnen sein, bevor \overrideTimeSignatureSettings aufgerufen wird. Das heißt, dass er entweder explizit begonnen wird oder sich Noten in dem Kontext befinden müssen, bevor \overrideTimeSignatureSettings aufgerufen wird:

\score {
  \relative c' {
    % This call will fail because the context isn't yet instantiated
    \overrideTimeSignatureSettings
    4/4 \timeSignatureFraction
    1/4 \baseMomentFraction
    3,1 \beatStructure
    #'() \beamExceptions
    \time 4/4
    c8 \markup{"Beamed (2 2)"}
    \repeat unfold 7 { c8 } |
    % This call will succeed
    \overrideTimeSignatureSettings
    4/4 \timeSignatureFraction
    1/4 \baseMomentFraction
    3,1 \beatStructure
    #'() \beamExceptions
    \time 4/4
    c8 \markup{"Beamed (3 1)"}
    \repeat unfold 7 { c8 } |
  }
}

Beamed (2 2) Beamed (3 1)
Veränderte Werte der Taktart-Eigenschaften können wieder auf den Standard zurückgesetzt werden:

\score{
\relative {
\repeat unfold 8 \{ c'8 \} |
\overrideTimeSignatureSettings 4/4 % timeSignatureFraction
1/4 % baseMomentFraction
3,1 % beatStructure
#'() % beamExceptions
\time 4/4
\repeat unfold 8 \{ c8 \} |
\revertTimeSignatureSettings 4/4
\time 4/4
\repeat unfold 8 \{ c8 \} |
}
}

Unterschiedliche Werte der Standard-Taktarteigenschaften für unterschiedliche Notensysteme können eingerichtet werden, indem man den Timing_translator und den Default_bar_line_engraver aus dem Score-Kontext in den Staff-Kontext verschiebt.

\score {
\new StaffGroup <<
\new Staff {
\overrideTimeSignatureSettings 4/4 % timeSignatureFraction
1/4 % baseMomentFraction
3,1 % beatStructure
#'() % beamExceptions
\time 4/4
\repeat unfold 8 \{ c'8 \}
}
\new Staff {
\overrideTimeSignatureSettings 4/4 % timeSignatureFraction
1/4 % baseMomentFraction
1,3 % beatStructure
#'() % beamExceptions
\time 4/4
\repeat unfold 8 \{ c'8 \}
}
>>
\layout {
\context {
\Score
\remove "Timing_translator"
\remove "Default_bar_line_engraver"
}
Vordefinierte Befehle
\numericTimeSignature, \defaultTimeSignature.

Ausgewählte Schnipsel

Time signature printing only the numerator as a number (instead of the fraction)

Sometimes, a time signature should not print the whole fraction (e.g. 7/4), but only the numerator (7 in this case). This can be easily done by using \override Staff.TimeSignature.style = #'single-digit to change the style permanently. By using \revert Staff.TimeSignature.style, this setting can be reversed. To apply the single-digit style to only one time signature, use the \override command and prefix it with a \once.

\relative c'' {  
\time 3/4  
c4 c c  
% Change the style permanently  
\override Staff.TimeSignature.style = #'single-digit  
\time 2/4  
c4 c  
\time 3/4  
c4 c c  
% Revert to default style:  
\revert Staff.TimeSignature.style  
\time 2/4  
c4 c  
% single-digit style only for the next time signature  
\once \override Staff.TimeSignature.style = #'single-digit  
\time 5/4  
c4 c c c c  
\time 2/4  
c4 c  
}
Siehe auch

Glossar: Abschnitt “time signature” in Glossar

Notationsreferenz: [Mensurale Taktartenbezeichnungen], Seite 415, [Verwaltung der Zeiteinheiten], Seite 111.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.


Metronomangabe

Eine Metronomanweisung wird wie folgt erstellt:

```
\relative {
  \tempo 4 = 120
  c'2 d
  e4. d8 c2
}
```

Metronombezeichnungen können auch für einen Zahlenbereich notiert werden:

```
\relative {
  \tempo 4 = 40 - 46
  c'4. e8 a4 g
  b,2 d4 r
}
```

Anstelle dessen kann auch Text als Argument angegeben werden:

```
\relative {
  \tempo "Allegretto"
  c''4 e d c
  b4. a16 b c4 r4
}
```

Wenn eine Metronombezeichnung und Text kombiniert wird, wird die Metronombezeichnung automatisch in Klammern gesetzt:

```
\relative {
  \tempo "Allegro" 4 = 160
  g'4 c d e
  d4 b g2
}
```
Kapitel 1: Musikalische Notation

Der Text kann ein beliebiges Textbeschriftungsobjekt sein:
\relative {
\tempo \markup { \italic Faster } 8 = 132
a'8-. r8 b-. r gis-. r a-. r
}

Eine Metronombezeichnung in Klammern ohne Text kann erstellt werden, indem eine leere Zeichenkette hinzugefügt wird:
\relative {
\tempo "" 8 = 96
d''4 g e c
}

Ausgewählte Schnipsel

Metronom- und Übungszeichen unter das System setzen
Normalerweise werden Metronom- und Übungszeichen über dem Notensystem ausgegeben. Um sie unter das System zu setzen, muss die direction-Eigenschaft von MetronomeMark oder RehearsalMark entsprechend verändert werden.
\layout {
indent = 0
ragged-right = ##f
}

% Metronome marks below the staff
\override Score.MetronomeMark.direction = #DOWN
\tempo 8. = 120
c''1

% Rehearsal marks below the staff
\override Score.RehearsalMark.direction = #DOWN
\mark \default
c''1
Das Tempo ohne Metronom-Angabe verändern

Um das Tempo für die MIDI-Ausgabe zu ändern, ohne eine Tempoangabe in den Noten auszugeben, kann die Metronombezeichnung unsichtbar gemacht werden:

\score {
\new Staff \relative c' {
\tempo 4 = 160
\set Score.tempoHideNote = ##t
\tempo 4 = 96
\d,4 fis a cis
d4 cis e d
}
\layout { }
\midi { }
}

Eine Metronombezeichnung als Textbeschriftung erstellen

Neue Metronombezeichnungen können als Textbeschriftung erstellt werden, aber sie ändern nicht das Tempo für die MIDI-Ausgabe.

\relative c' {
\tempo \markup {
\concat {
\smaller \general-align #Y #DOWN \note "16." #1
" = 
\smaller \general-align #Y #DOWN \note "8" #1
}
}
\c1
\c4 \c' \c,2
}

Zu Einzelheiten siehe Abschnitt 1.8.2 [Text formatieren], Seite 223.
Siehe auch


Notationsreferenz: Abschnitt 1.8.2 [Text formatieren], Seite 223, Abschnitt 3.5 [MIDI-Ausgabe], Seite 487.

Schnipsel: Abschnitt “Staff notation” in Schnipsel.

Referenz der Interna: Abschnitt “MetronomeMark” in Referenz der Interna.

Auftakte

Verkleinerte Takte, wie etwa ein Auftakt, werden mit dem Befehl \partial notiert, dessen Syntax lautet:

\partial \text{Dauer}

wobei \text{Dauer} die Länge der Noten darstellt, bevor der nächste vollständige Takt beginnt:

\begin{verbatim}
\relative {
  \time 3/4
  \partial 8
  e8 | a4 c8 b c4 |
}
\end{verbatim}

Die \text{Dauer} kann ein beliebiger Wert kleiner als der vollständige Takt sein.

\begin{verbatim}
\relative {
  \time 3/4
  \partial 4.
  r4 e'8 | a4 c8 b c4 |
}
\end{verbatim}

\partial \text{Dauer} kann auch folgendermaßen geschrieben werden:

\set Timing.measurePosition \text{-Länge der Dauer}

So wird etwa aus \partial 8:

\begin{verbatim}
\time 3/4
\set Timing.measurePosition = #(ly:make-moment -1/8)
\partial 8 | a4 c8 b c4 |
\end{verbatim}

Die Eigenschaft \text{measurePosition} (Takt-Position) enthält eine rationale Zahl, die anzeigt, wie groß der Abstand zum Taktanfang ist. Deshalb ist sie eine negative Zahl; \partial 4 wird also intern übersetzt zu \text{-4} was soviel bedeutet wie: „Eine Viertel bleibt übrig vom ganzen Takt.“
Kapitel 1: Musikalische Notation

Siehe auch

Glossar: Abschnitt “anacrusis” in Glossar.
Notationsreferenz: [Verzierungen], Seite 105.
Schnipsel: Abschnitt “Rhythms” in Schnipsel.

Bekannte Probleme und Warnungen

\partial ist nur für den Anfang eines Stückes vorgesehen. Wenn der Befehl nach dem Anfang verwendet wird, können Warnungen oder Probleme auftreten. In solchem Fall sollten Sie \set Timing.measurePosition benutzen.

\time 6/8
\partial 8
e8 | a4 c8 [ c b] |
\set Timing.measurePosition = #(ly:make-moment -1/4)
r8 e,8 | a4 c8 [ c b] |

Musik ohne Metrum

In Musik mit Metrum werden Taktstriche automatisch eingefügt und Taktzahlen automatisch berechnet. In Musik ohne Metrum hingegen (etwa Kadenzen) ist das nicht gewollt und kann „ausgeschaltet“ werden mit dem Befehl \cadenzaOn, um dann wieder zum normalen Verhalten mit \cadenzaOff zurückzukehren.

\relative c'' {c4 d e d \cadenzaOn c4 c d8[ d d] f4 g4. \cadenzaOff \bar "|"
d4 e d c}

Taktnummerierung wird am Ende der Kadenz wieder aufgenommen:

% Show all bar numbers
\override Score.BarNumber.break-visibility = #all-visible
c4 d e d \cadenzaOn c4 c d8[ d d] f4 g4. \cadenzaOff \bar "|"
d4 e d c
Ein neuer Takt wird nie innerhalb der Kadenz begonnen, auch wenn einer oder mehrere \bar-Befehle eingefügt werden. Darum müssen auch Erinnerungsversetzungssignale manuell eingefügt werden. Siehe [Versetzungssignale], Seite 5.

```latex
\begin{verbatim}
c4 d e d \\cadenzAOn
cis4 d cis d \\bar "|"
cis4 d cis! d \\cadenzaOff \\bar "|
\end{verbatim}
```

Automatische Bebalkung wird durch \cadenzaOn ausgestellt. Darum müssen alle Balken in Kadenz manuell eingegeben werden (siehe [Manuelle Balken], Seite 88).

```latex
\begin{verbatim}
\relative {
  \repeat unfold 8 { c''8 }
  \cadenzAOn 
cis8 c c c c \\bar"|"
c8 c c \\cadenzaOff 
  \repeat unfold 8 { c8 }
}
\end{verbatim}
```

Diese vordefinierten Befehle wirken sich auf alle Systeme in der Partitur aus, auch wenn sie nur in einer einzigen Stimme notiert werden. Um dieses Verhalten zu ändern, müssen Sie `Timing_translator` aus dem `Score`-Kontext in den `Staff`-Kontext verschieben, wie gezeigt in [Polymetrische Notation], Seite 71.

**Vordefinierte Befehle**

\cadenzAOn, \cadenzaOff.

**Siehe auch**

Glossar: Abschnitt “cadenza” in Glossar.

Notationsreferenz: Abschnitt 5.4.6 [Sichtbarkeit von Objekten], Seite 589, [Polymetrische Notation], Seite 71, [Manuelle Balken], Seite 88, [Versetzungssignale], Seite 5.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.

**Bekannte Probleme und Warnungen**

Automatische Zeilen- und Seitenumbrüche werden nur an Taktstrichen vorgenommen, sodass „unsichtbare“ Taktstriche manuell eingefügt werden müssen, damit Umbrüche erlaubt werden:

\bar ""
Man muss explizit einen Voice-Kontext erstellen, wenn man ein Stück mit \texttt{cadenzaOn} beginnen will, weil sonst ein seltsamer Fehler auftreten kann.

\begin{verbatim}
\new Voice {
  \relative c' {
    \cadenzaOn
    c16[^"Solo Free Time" d e f] g2.
    \bar "||"
    \cadenzaOff
  }
}
\end{verbatim}

\section*{Polymetrische Notation}

Polymetrische Notation ist direkt unterstützt, oder indem man das sichtbare Taktart-Symbol verändert und zusätzlich die Notendauern skaliert.

\section*{Unterschiedliche Taktarten mit gleicher Taklänge}

Für jedes System wird eine gemeinsame Taktart gesetzt und dann die Einstellung von \texttt{timeSignatureFraction} auf den gewünschten Bruch geändert. Mit dem Befehl \texttt{scaleDurations} werden die Dauern der Noten auf jedem System dann auf die gemeinsame Taktart skaliert. Siehe auch [Tondauern skalieren], Seite 48.


\begin{verbatim}
\relative <<
  \new Staff {
    \time 3/4
    c'4 c c |
    c4 c c |
  }
  \new Staff {
    \time 3/4
    \set Staff.timeSignatureFraction = 9/8
    \scaleDurations 2/3
    \repeat unfold 6 { c8[ c c] }
  }
  \new Staff {
    \time 3/4
    \set Staff.timeSignatureFraction = 10/8
    \scaleDurations 3/5 {
      \repeat unfold 2 { c8[ c c] }
      \repeat unfold 2 { c8[ c ] } |
      c4. c4. \tuplet 3/2 { c8[ c c] } c4
    }
  }
}
\end{verbatim}
Unterschiedlichen Taktarten mit unterschiedlicher Taktlänge

Jedes System kann auch eine eigene unabhängige Taktart erhalten. Dazu muss der \texttt{Timing\_translator} und der \texttt{Default\_bar\_line\_engraver} in den \texttt{Staff}-Kontext verschoben werden.

\begin{verbatim}
\layout {
  \context {
    \Score
    \remove "Timing\_translator"
    \remove "Default\_bar\_line\_engraver"
  }
  \context {
    \Staff
    \consists "Timing\_translator"
    \consists "Default\_bar\_line\_engraver"
  }
}

% Now each staff has its own time signature.

\relative <<
  \new Staff {
    \time 3/4
    c'4 c c |
    c4 c c |
  }

  \new Staff {
    \time 2/4
    c4 c |
    c4 c |
    c4 c |
  }

  \new Staff {
    \time 3/8
    c4. |
    c8 c c |
    c4. |
    c8 c c |
  }
\end{verbatim}
Zusammengesetzte Taktarten

Taktarten aus mehreren Teilen werden mit der Funktion \compoundMeter erstellt. Die Syntax ist folgende:

\compoundMeter #'(Liste aus Listen)

Die einfachste Konstruktion ist eine einzige Liste, wobei die letzte Zahl den Nenner des Bruches darstellt, während die vorherkommenden Zahlen die Zähler sind.

\relative {
  \compoundMeter #'((2 2 2 8))
  \repeat unfold 6 c'8 \repeat unfold 12 c16
}


\relative {
  \compoundMeter #'((1 4) (3 8))
  \repeat unfold 5 c'8 \repeat unfold 10 c16
}

\relative {
  \compoundMeter #'((1 2 3 8) (3 4))
  \repeat unfold 12 c'8
}

Siehe auch


Notationsreferenz: [Taktangabe], Seite 60, [Automatische Balken], Seite 77, [Manuelle Balken], Seite 88, [Tondauern skalieren], Seite 48.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.

Bekannte Probleme und Warnungen
Wenn unterschiedliche Taktarten parallel benutzt werden, werden Noten auf demselben musikalischen Moment horizontal auf die gleiche Position gesetzt. Die unterschiedlichen Taktlinien führen allerdings dazu, dass die Noten nicht ganz so regelmäßig gesetzt werden, wie es ohne unterschiedliche Taktarten der Fall wäre.

Automatische Aufteilung von Noten
Lange Noten, die über Taktlinien hinüberreichen, können automatisch in übergebundene Noten aufgeteilt werden. Dieses Verhalten erreicht man, indem der Abschnitt “Note_heads_engraver” in Referenz der Interna mit dem Abschnitt “Completion_heads_engraver” in Referenz der Interna ausgetauscht wird. Auf gleiche Art können lange Pausen, die über Taktgrenzen reichen, automatisch aufgeteilt werden, indem man den Rest_engraver mit dem Completion_rest_engraver ersetzt. Im nächsten Beispiel werden Noten und Pausen, die über die Taktlinie dauern, aufgeteilt; Noten werden auch übergebunden.

\new Voice \with {\remove "Note_heads_engraver" \consists "Completion_heads_engraver" \remove "Rest_engraver" \consists "Completion_rest_engraver" } \relative {\c'2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2 r1*2}  

Dieser Engraver teilt alle Noten und Pausen auf, die über eine Taktlinie dauern und fügt für Noten Bindebogen hinzu. Er kann unter Anderem dann nützlich sein, wenn man kompliziert gestaltete 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.

Siehe auch
Glossar: Abschnitt “tie” in Glossar
Schnipsel: Abschnitt “Rhythms” in Schnipsel.

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.

Melodierhythmus anzeigen

\new RhythmicStaff {
  \new Voice = "myRhythm" \relative {
    \time 4/4
    c'4 e8 f g2
    r4 g g f
    g1
  }
}
\new Lyrics {
  \lyricsto "myRhythm" {
    This is my song
    I like to sing
  }
}

This is my song I like to sing

Akkordnotation für Gitarren bezeichnet auch oft zusätzlich den geschlagenen Rhythmus. Das kann notiert werden unter Verwendung des Pitch_squash_engraver und indem Tonhöhenimprovisation eingeschaltet wird mit \improvisationOn.

\new ChordNames {
  \chordmode {
    c1 f g c
  }
}
\new Voice \with {
  \consists "Pitch_squash_engraver"
} \relative c' {
  \improvisationOn
  c4 c8 c c4 c8 c
  f4 f8 f f4 f8 f
  g4 g8 g g4 g8 g
  c4 c8 c c4 c8 c
}

This is my song I like to sing C F G C
Vordefinierte Befehle

`\improvisationOn, \improvisationOff`.

Ausgewählte Schnipsel

*Schlagrhythmus für Gitarren*

In Gitarrennotation kann neben Melodie, Akkordbezeichnungen und Bunddiagrammen auch der Schlagrhythmus angegeben werden.

```latex
\include "predefined-guitar-fretboards.ly"
<<
\new ChordNames {
  \chordmode {
    c1 | f | g | c
  }
}
\new FretBoards {
  \chordmode {
    c1 | f | g | c
  }
}
\new Voice \with {
  \consists "Pitch_squash_engraver"
} {
  \relative c' {\improvisationOn
  c4 c8 c4 c8 c
  f4 f8 f4 f4 f8 f
  g4 g8 g4 g8 g
  c4 c8 c4 c8 c
  }
}
\new Voice = "melody" {
  \relative c' {\improvisationOn
  c2 e4 e4
  f2. r4
  g2. a4
  e4 c2.
  }
}
\new Lyrics {
  \lyricsto "melody" {
  This is my song.
  I like to sing.
  }
  }
>>
```
This is my song. I like to sing.

Siehe auch
Schnipsel: Abschnitt “Rhythms” in Schnipsel.
Referenz der Interna: Abschnitt “RhythmicStaff” in Referenz der Interna, Abschnitt “Pitch_squash_engraver” in Referenz der Interna.

1.2.4 Balken

Automatische Balken
LilyPond setzt Balken (engl. beam) automatisch.

```
\relative c'' {
  \time 2/4 c8 c c c
  \time 6/8 c8 c c c8. c16 c8
}
```

Wenn diese automatischen Entscheidungen nicht gut genug sind, können die Balken auch explizit eingegeben werden, siehe [Manuelle Balken], Seite 88. Balken müssen auch auf diese Weise eingegeben werden, wenn sie über Pausen hinwegreichen sollen.

Wenn automatische Bebalkung nicht benötigt wird, kann sie mit dem Befehl \autoBeamOff aufgehoben werden und mit dem Befehl \autoBeamOn wieder eingeschaltet werden.

```
\relative c' {
  c4 c8 c8. c16 c8. c16 c8
  \autoBeamOff
c4 c8 c8. c16 c8.
  \autoBeamOn
c16 c8
```
Kapitel 1: Musikalische Notation

Achtung: Wenn Balken eingesetzt werden, um Melismen in Gesang zu notieren, sollte die automatische Bebalkung mit `\autoBeamOff` ausgeschaltet werden und die Balken manuell notiert werden. Die Benutzung von `\partcombine` zusammen mit `\autoBeamOff` kann zu unbeabsichtigten Ergebnissen führen. Siehe die Schnipsel für mehr Information.

Balkenmuster, die sich von den automatisch erstellen unterscheiden, können erstellt werden, siehe [Einstellung von automatischen Balken], Seite 80.

Vordefinierte Befehle
`\autoBeamOff, \autoBeamOn`.

Ausgewählte Schnipsel

**Balken über Zeilenumbrüche**
Zeilenumbrüche sind normalerweise während Balken verboten. Das kann geändert werden.

```
\relative c' {  
  \override Beam.breakable = ##t
  c8 c[c] c[c] c[c] c\break
  c8] c[c] c[c] c[c] c}
```

**Balken für weit auseinander liegende Noten ändern**

```
{  
f8 f''8 f8 f''8
  \override Beam.auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```
Partcombine und autoBeamOff

Die Funktionsweise von \autobeamoff, wenn es zusammen mit \partcombine eingesetzt wird, kann schwer zu verstehen sein. Es kann besser sein, anstatt dessen \set Staff.autobeaming = ##f zu benutzen, um sicherzustellen, dass die automatische Bebalkung für das gesamte System ausgeschaltet ist.

\partcombine funktioniert offensichtlich mit 3 Stimme (Hals nach oben einfach, Hals nach unten einfach, Hals nach oben kombiniert).

Ein \autobeamoff-Befehl im ersten Argument von \partcombine gilt für die Stimme, die zu dem Zeitpunkt aktiv ist, an dem der Befehl verarbeitet wird, entweder für Hals nach oben, nach unten oder Hals nach oben kombiniert. Ein \autobeamoff-Befehl im zweiten Argument gilt für die Stimme, die mit Hals nach unten einfach ist.

Um \autobeamoff zu benutzen, damit alle automatischen Balken aufhören, wenn man es mit \partcombine verwendet, muss \autobeamoff dreimal aufgerufen werden.

\{
% \set Staff.autoBeaming = ##f % turns off all autobeaming
\partcombine
{
  \autobeamoff % applies to split up stems
  \repeat unfold 4 a'16
  % \autobeamoff % applies to combined up stems
  \repeat unfold 4 a'8
  \repeat unfold 4 a'16
}
{
  \autobeamoff % applies to down stems
  \repeat unfold 4 f'8
  \repeat unfold 8 f'16 |
}
\}

Siehe auch

Notationsreferenz: [Manuelle Balken], Seite 88, [Einstellung von automatischen Balken], Seite 80.

Installierte Dateien: scm/auto-beam.scm.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.

Bekannte Probleme und Warnungen

Die Eigenschaften eines Balkens werden am Beginn seiner Konstruktion bestimmt. Alle zusätzlichen Änderungen der Balkeneigenschaften, die auftreten, bevor der Balken zuende ist, werden nicht ausgewertet, bevor nicht der nächste Balken beginnt.

Einstellung von automatischen Balken


Standardmäßig sind beamExceptions-Regel für alle häufigen Taktarten vordefiniert, sodass die beamExceptions-Regeln deaktiviert werden müssen, wenn die Balkenplatzierung aufgrund von baseMoment und beatStructure stattfinden soll. Die beamExceptions-Regeln werden deaktiviert durch

\set Timing.beamExceptions = #'()  

Bebalkung basierend auf baseMoment und beatStructure

In den meisten Fällen enden automatische Balken am Ende eines Taktes. Die Endpunkte für Schläge werden durch die Kontexteigenschaften baseMoment und beatStructure bestimmt.  


\time 5/16  
c16"default" c c c c |  
\set Timing.beatStructure = 2,3  
c16"(2+3)" c c c c |  
\set Timing.beatStructure = 3,2  
c16"(3+2)" c c c c |  

\time 4/4  
a8"default" a a a a a a a  

Wenn eine häufige Taktart benutzt wird, muss beamExceptions deaktiviert werden, damit beatStructure funktionieren kann. Der \set Timing.beamExceptions = '#'()-Befehl kann immer eingefügt werden, wenn die Bebalkung von beatStructure bestimmt werden soll.

\time 4/4  
a8"default" a a a a a a a  

\set Timing.baseMoment = #(ly:make-moment 1/4)  
\set Timing.beatStructure = 1,1,1,1  
a8"no change" a a a a a a a  

\set Timing.beamExceptions = '#'()  
\set Timing.baseMoment = #(ly:make-moment 1/4)  
\set Timing.beatStructure = 1,1,1,1  
a8"changed" a a a a a a a
Balkenregelveränderungen können auf bestimmte Kontexte beschränkt werden. Wenn keine Regeln in einen unteren Kontext definiert sind, gelten die Regeln des höheren Kontext, in dem sich der niedrigere befindet.

\new Staff <<
\time 7/8
\set Staff.beatStructure = 2,3,2
\new Voice = one {
   \relative {
      a'8 a a a a a a
   }
}
\new Voice = two {
   \relative {
      \voiceTwo
      \set Voice.beatStructure = 1,3,3
      f'8 f f f f f f
   }
}
>>

Wenn mehrere Stimmen eingesetzt werden, muss der Staff-Kontext definiert werden, wenn die Balkenregeln auf alle Stimmen des Systems angewendet werden sollen:

\time 7/8
% rhythm 3-1-1-2
% Context applied to Voice by dafault -- does not work correctly
% Because of autogenerated voices, all beating will
% be at baseMoment (1 . 8)
\set Timing.baseMoment = #(ly:make-moment 1/16)
\set Timing.beatStructure = 7,3
<< {a8 a a a16 a a a a8 a} \{f4. f8 f f f} >>

% Works correctly with context Staff specified
\set Staff.beatStructure = 3,1,1,2
<< {a8 a a a16 a a a a8 a} \{f4. f8 f f f} >>

Der Wert von baseMoment kann angepasst werden, um das Bebalkungsverhalten zu ändern, wenn gewünscht. In diesem Fall muss der Wert von beatStructure so gesetzt werden, dass er kompatibel mit dem neuen Wert von baseMoment ist.

\time 5/8
\set Timing.baseMoment = #(ly:make-moment 1/16)
\set Timing.beatStructure = 7,3
\repeat unfold 10 { a16 }
Kapitel 1: Musikalische Notation


**Bebalkung basierend auf beamExceptions**

Besondere automatische Bebalkungsregeln (außer dass ein Balken auf einem Taktschlag aufhört) sind in der `beamExceptions`-Eigenschaft definiert.

```
\time 3/16
\set Timing.beatStructure = 2,1
\set Timing.beamExceptions =
  #'( ; start of alist
    (end . ; entry for end of beams
      ( ; start of alist of end points
        ((1 . 32) . (2 2 2)) ; rule for 1/32 beams -- end each 1/16
      )))
    % close all entries

c16 c c |
\repeat unfold 6 { c32 } |
```

**beamExceptions** ist eine Aliste mit einem Schlüssel der Regeltypen (rule-type) und einem Wert der Bebalkungsregeln (beaming-rules).

Im Moment ist der einzige mögliche rule-type ’end für ein Balkenende.

Beaming-rules ist eine Scheme-Aliste (oder eine paarige Liste), die den Balkentyp und die Gruppierung anzeigt, die auf Balken angewendet werden, welche Noten mit einer kürzesten Dauer des Balkentyps enthalten.

```scheme
#'((beam-type1 . grouping-1)
  (beam-type2 . grouping-2)
  (beam-type3 . grouping-3))
```

Beam-type ist ein Scheme-Paar, das die Dauer eines Balkens anzeigt, etwa `(1 . 16)` für ein Sechzehntel.

Grouping ist eine Scheme-Liste, die die auf den Balken anzuwendene Gruppierung anzeigt. Die Gruppierung wird in Einheiten des Balkentyps angegeben.

**Achtung:** Ein `beamExceptions`-Wert muss eine *vollständige* Ausnahme-Liste sein. Das heißt, dass jede Ausnahme, die angewendet werden soll, auch in die Einstellungen mit aufgenommen werden muss. Es ist nicht möglich, nur eine der Einstellungen zu ändern, zu entfernen oder hinzuzufügen. Das mag seltsam erscheinen, bedeutet aber, dass die aktuellen Balkenregeln bekannt sein müssen, um ein neues Bebalkungsmuster definieren zu können.

Wenn die Taktart geändert wird, werden neue Standardwerte für `Timing.baseMoment`, `Timing.beatStructure` und `Timing.beamExceptions` definiert. Wenn die Taktart definiert
wird, werden die automatischen Bebalkungsregeln für den *Timing*-Kontext auf den Standard zurückgesetzt.

\relative a' {
  \time 6/8
  \repeat unfold 6 { a8 }
  \% group (4 + 2)
  \set Timing.beatStructure = 4,2
  \repeat unfold 6 { a8 }
  \% go back to default behavior
  \time 6/8
  \repeat unfold 6 { a8 }
}\%

Die automatischen Standardeinstellungen für die Bebalkung einer Taktart werden in der Datei `scm/time-signature-settings.scm` bestimmt. Änderungen der automatischen Bebalkungsregeln für eine Taktart werden in [Taktangabe], Seite 60, beschrieben.

Die meisten automatischen Bebalkungsregeln für eine Taktart enthalten einen Eintrag für `beamExceptions`. Beispielsweise wird in einem 4/4-Takt versucht, den Takt in zwei Hälften zu teilen, wenn nur Achtelnoten vorkommen. Die `beamExceptions`-Regel kann die `beatStructure`-Einstellung überschreiben, wenn `beamExceptions` nicht zurückgesetzt wird:

\time 4/4
\set Timing.baseMoment = #(ly:make-moment 1/8)
\set Timing.beatStructure = 3,3,2
\% This won't beam (3 3 2) because of beamExceptions
\repeat unfold 8 {c''8} |
\% This will beam (3 3 2) because we clear beamExceptions
\set Timing.beamExceptions = #'()
\repeat unfold 8 {c''8}

Auf gleiche Art werden Achtelnoten im 3/4-Takt über den ganzen Takt hin mit Balken versehen. Damit Achtelnoten im 3/4-Takt auf jedem Schlag einen neuen Balken erhalten, muss `beamExceptions` verändert werden:

\time 3/4
\% by default we beam in (6) due to beamExceptions
\repeat unfold 6 {a8} |
\% This will beam (1 1 1) due to beatLength
\set Timing.beamExceptions = #'()
\repeat unfold 6 {a8}

\relative a' { 
  \time 3/4
  r4. a8 a a |
  \set Timing.beamHalfMeasure = ##f
  r4. a8 a a |
}

Wie die automatische Bebalkung funktioniert

Wenn die automatische Bebalkung aktiviert ist, wird die Platzierung der automatischen Balken durch die Kontexteigenschaften baseMoment, beatStructure und beamExceptions bestimmt.

Die folgenden Regeln, in der Reihenfolge ihrer Priorität, gelten, wenn das Aussehen der Balken bestimmt wird:

- Wenn ein manueller Balken mit [...] definiert ist, wird er gesetzt, andernfalls
- wenn eine Balkenendung-Regel für den Balkentyp in beamExceptions definiert ist, wird sie verwendet, um die gültigen Plätze für Balkenenden zu berechnen, andernfalls
- wenn eine Balkenendung-Regel für einen größeren Balkentyp in beamExceptions definiert ist, wird sie verwendet, um die gültigen Plätze für Balkenenden zu berechnen, andernfalls
- benutze die Werte von baseMoment und beatStructure, um die Enden der Balken im Takt zu definieren und beende Balken am Ende jedes Taktes.

In den oben genannten Regeln ist der Balkentyp die Dauer der kürzesten Note der bebalkten Gruppe.

Die Standardbebalkungsregeln finden sich in der Datei scm/time-signature-settings.scm.

Ausgewählte Schnipsel

Subdividing beams

The beams of consecutive 16th (or shorter) notes are, by default, not subdivided. That is, the three (or more) beams stretch unbroken over entire groups of notes. This behavior can be modified to subdivide the beams into sub-groups by setting the property subdivideBeams. When set, multiple beams will be subdivided at intervals defined by the current value of baseMoment by reducing the multiple beams to the number of beams that indicates the metric value of the subdivision. If the group following the division is shorter than the current metric value (usually because the beam is incomplete) the number of beams reflects the longest possible subdivision group. However, if there is only one note left after the division this restriction isn’t applied. Note that baseMoment defaults to one over the denominator of the current time signature if not set explicitly. It must be set to a fraction giving the duration of the beam sub-group using the ly:make-moment function, as shown in this snippet. Also, when baseMoment is changed, beatStructure should also be changed to match the new baseMoment:

\relative c' { 
  c32[ c c c c c c c]
  \set subdivideBeams = ##t
Bebalkung nach Taktschlag

Sekundäre Balken können in die Richtung gesetzt werden, die ihrer rhythmischen Zugehörigkeit entspricht. Der erste Balken ist zusammengefasst (Standard), der zweite Sechzehntelbalken zeigt den Taktschlag an.

\relative c' {  
\time 6/8  
a8. a16 a a  
\set strictBeatBeaming = ##t  
a8. a16 a a  
}

Dirigierzeichen, Taktgruppenzeichen

Optionen, mit denen die Balken in einem Takt gruppiert werden, sind durch die Scheme-Funktion \set-time-signature erhältlich, die drei Argumente braucht: Die Zahl der Taktschläge, die Länge des Schlages und die interne gruppiert von Balken in dem Takt.

```plaintext
\score {
  \new Voice \relative c'' {
    \time 9/8
    g8 g d d g g a( bes g) |
    \set Timing.beatStructure = 2,2,2,3
    g8 g d d g g a( bes g) |
    \time 4,5 9/8
    g8 g d d g g a( bes g) |
    \time 5/8
    a4. g4 |
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```

*Balkenenden auf Score-Ebene*

Balkenenderegeln, die im Score-Kontext definiert werden, wirken sich auf alle Systeme aus, können aber auf Staff- und Voice-Ebene neu verändert werden:

```plaintext
\relative c' ' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.baseMoment = #(ly:make-moment 1/8)
  \set Score.beatStructure = 3,4,3
  %%%
  \new Staff {
    c8 c c c c c c c
  }
  \new Staff {
    % Modify beaming for just this staff
    \set Staff.beatStructure = 6,4
    c8 c c c c c c c
  }
  \new Staff {
```

```plaintext
```
Kapitel 1: Musikalische Notation

% Inherit beaming from Score context
<<
{,
\voiceOne
c8 c c c c c c c c
}
% Modify beaming for this voice only
\new Voice {
\voiceTwo
\set Voice.beatStructure = 6,4
a8 a a a a a a a a
}
>>
}
>

Siehe auch

Installierte Dateien: scm/beam-settings.scm.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.


Bekannte Probleme und Warnungen

Wenn eine Partitur endet, während ein automatischer Balken noch nicht beendet wurde und weiterhin Noten erwartet, wird dieser letzte Balken nicht ausgegeben. Das Gleiche gilt auch für polyphone Stimmen, die mit der << ... \ ... >>-Konstruktion notiert wurden. Wenn eine polyphone Stimme endet, während ein Balken noch weitere Noten erwartet, wird der Balken nicht gesetzt. Eine Notlösung für dieses Problem ist, den letzten Balken in der Stimme oder Partitur manuell zu setzen.


<<
\new Staff {
\time 3/4
Kapitel 1: Musikalische Notation

\set Timing.baseMoment = #(ly:make-moment 1/8)
\set Timing.beatStructure = 1,5
\repeat unfold 6 { a8 }

}\new Staff {
\repeat unfold 6 { a8 }
}

>>

\new Staff {
\overrideTimeSignatureSettings
3/4 % timeSignatureFraction
1/8 % baseMomentFraction
1,5 % beatStructure
#'() % beamExceptions
\time 3/4
\repeat unfold 6 { a'8 }
}
\new Staff {
\time 3/4
\repeat unfold 6 { a'8 }
}
\new Staff {
\relative { r4 r8[ g' a r] r g[ | a] r }

Die Standardbebalkungsregeln für die Taktart können auch verändert werden, sodass die gewünschte Bemerkung immer benutzt wird. Veränderungen der automatischen Bemerkungsregeln für eine Taktart sind in [Taktangabe], Seite 60, beschrieben.

<<
\new Staff {
\overrideTimeSignatureSettings
3/4 % timeSignatureFraction
1/8 % baseMomentFraction
1,5 % beatStructure
#'() % beamExceptions
\time 3/4
\repeat unfold 6 { a'8 }
}
\new Staff {
\time 3/4
\repeat unfold 6 { a'8 }
}

>>

Manuelle Balken


\relative { r4 r8[ g' a r] r g[ | a] r }
Die Richtung von Balken kann mit den Richtungszeichen verändert werden:
\relative { c''8\^[ d e] c,\_[ d e f g] }

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

\relative {
c''4 d8[
grace { e32 d c d }
o8] e\[ e 
grace { f16 }
o8 e]
}

Noch bessere manuelle Kontrolle über die Balken kann durch Setzen der Eigenschaften stemLeftBeamCount und stemRightBeamCount erreicht werden. Sie bestimmen die Anzahl von Balken, die rechts und links vom Hals der nächsten Note gesetzt werden sollen. Wenn eine Eigenschaft gesetzt ist, wird ihr Wert nur einmal eingesetzt und dann wieder auf Null gesetzt. Im folgenden Beispiel hat das letzte f nur einen Balken an seiner linken Seite (der als Achtelbalken der gesamten Gruppe gewertet wird).
\relative a' {
a8[ r16 f g a]
a8[ r16 
\set stemLeftBeamCount = #2
\set stemRightBeamCount = #1
f16 
\set stemLeftBeamCount = #1
g16 a]
}
Ausgewählte Schnipsel

Gerade Fähnchen und überstehende Balkenenden

Gerade Fähnchen an einzelnen Noten und überstehende Balkenenden bei bebalkten Notengruppen sind möglich mit einer Kombination aus stemLeftBeamCount, stemRightBeamCount und Paaren von [-]-Balkenbegrenzungen.

Für gerade Fähnchen, die nach rechts zeigen, kann [] eingesetzt werden und stemLeftBeamCount auf Null gesetzt werden (wie Bsp. 1).

Für gerade Fähnchen, die nach links zeigen, muss stemRightBeamCount eingesetzt werden (Bsp. 2).

Für überstehende Balkenenden nach rechts muss stemRightBeamCount auf einen positiven Wert gesetzt werden, für Balkenenden, die nach links zeigen benutzt man stemLeftBeamCount (Bsp. 3).

Manchmal können einzelne Noten, die von Pausen umgeben sind, auch Balkenenden in beide Richtungen tragen. Das geschieht mit [-]-Klammern (Bsp. 4).

(%set stemLeftBeamCount entspricht immer dem Befehl \once \set. Anders gesagt müssen die Einstellungen immer wieder wiederholt werden und die Fähnchen des letzten Sechzehntels im letzten Beispiel haben nichts mit dem \set-Befehl zwei Noten vorher zu tun.)

\score {
<<
% Example 1
\new RhythmicStaff {
  \set stemLeftBeamCount = #0
  c16[]
  r8.
}
% Example 2
\new RhythmicStaff {
  r8.
  \set stemRightBeamCount = #0
  16[]
}
% Example 3
\new RhythmicStaff {
  16 16
  \set stemRightBeamCount = #2
  16 r r
  \set stemLeftBeamCount = #2
  16 16 16
}
% Example 4
\new RhythmicStaff {
  16 16
  \set stemRightBeamCount = #2
  16 r16
  16[]
  r16
  \set stemLeftBeamCount = #2
  16 16
}
>>
Siehe auch

Notationsreferenz: Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581, [Verzierungen], Seite 105.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.


Gespreizte Balken


Wenn die Anordnung der Noten und die MIDI-Ausgabe das Ritardando oder Accelerando, wie es die Spreizung angibt, reflektieren soll, müssen die Noten als ein musikalischer Ausdruck notiert werden, der von geschweiften Klammern umgeben ist und dem ein featheredDurations-(gespreizteDauern)-Befehl vorangestellt ist, der das Verhältnis der ersten und letzten Dauer definiert.

Die eckigen Klammern geben die Reichweite des Balkens an und die geschweiften Klammern zeigen, auf welche Noten sich die Veränderung der Dauern auswirkt. Normalerweise bezieht sich das auf die selbe Notengruppe, aber das ist nicht unbedingt erforderlich: beide Befehle sind unabhängig voneinander.

Im folgenden Beispiel nehmen die acht 16-Noten exakt die gleiche Zeit ein wie eine halbe Note, aber die erste Note ist halb so lang wie die letzte der Gruppe, und die Noten dazwischen werden stufenweise verlängert. Die ersten vier 32-Noten beschleunigen stufenweise das Tempo, während die darauffolgenden vier 32-Noten ein gleichmäßiges Tempo haben.

\relative c' {
  \override Beam.grow-direction = #LEFT
  \featherDurations #(ly:make-moment 2/1)
  { c16[ c c c c c c c ] }
  \override Beam.grow-direction = #RIGHT
  \featherDurations #(ly:make-moment 2/3)
  { c32[ d e f ] }
Kapitel 1: Musikalische Notation

Die Platzierung der Noten im Druckbild entspricht den Notendauern nur annähernd, aber die MIDI-Ausgabe ist exakt.

Vordefinierte Befehle
\featherDurations.

Siehe auch
Snippets: Abschnitt “Rhythms” in Schnipsel.

Bekannte Probleme und Warnungen
Der \featherDurations-Befehl funktioniert nur mit kurzen Notenabschnitten, und wenn die Zahlen in den Brüchen klein sind.

1.2.5 Takte

Taktstriche
Taktstriche trennen Takte voneinander, werden aber auch verwendet, um Wiederholungen anzuzeigen. Normalerweise werden sie automatisch nach Vorgabe der aktuellen Taktart eingefügt.

Die einfachen, automatisch eingefügten Taktstriche können mit dem \bar-Befehl geändert werden. Eine doppelter Taktstrich etwa wird normalerweise am Ende eines Stückes gesetzt:
\relative { e'4 d c2 \bar "|." }

Es ist kein Fehler, wenn die letzte Note in einem Takt nicht zum automatisch eingefügten Taktstrich aufhört: es wird angenommen, dass die Note im nächsten Takt weitergeht. Wenn aber eine ganze Reihe solcher überlappenden Takte auftritt, können die Noten gedrungen aussehen oder sogar über den Seitenrand hinausragen. Das kommt daher, dass Zeilenumbrüche nur dann vorgenommen werden, wenn ein vollständiger Takt auftritt, also ein Takt, an dem alle Noten vor dem Taktstrich zu Ende sind.

**Achtung:** Eine falsche Dauer kann bewirken, dass Zeilenumbrüche verhindert werden, woraus resultiert, dass die Noten entweder sehr stark gedrängt auf der Zeile notiert werden, oder die Zeile über den Seitenrand hinausragt.

Zeilenumbrüche werden erlaubt, wenn ein Taktstrich manuell eingefügt wird, auch, wenn es sich um keinen vollständigen Takt handelt. Um einen Zeilenumbruch zu erlauben, ohne den Taktstrich auszugeben, kann
\bar ""
benutzt werden. Damit wird ein unsichtbarer Taktstrich an dieser Stelle eingefügt und damit ein Zeilenumbruch erlaubt (aber nicht erzwungen), ohne dass sich die Anzahl der Takte erhöhen würde. Um einen Zeilenumbruch zu erzwingen, siehe Abschnitt 4.3.1 [Zeilenumbrüche], Seite 512.


Dabe gilt zu beachten, dass manuell gesetzten Taktstriche nur visuell sichtbar sind. Sie wirken sich auf keine der Eigenschaften aus, die ein normaler Taktstrich beeinflussen würde, wie etwa Taktzahlen, Versetzungszeichen, Zeilenumbrüche usw. Sie beeinflussen auch nicht die Berechnung und Platzierung von weiteren automatischen Taktstrichen. Wenn ein manueller Taktstrich dort gesetzt wird, wo ein automatischer Taktstrich sowieso wäre, werden die Auswirkungen des originalen Taktstriches nicht verändert.

Manuell können zwei einfache Taktstriche und zusätzlich fünf Arten eines doppelten Taktstriches gesetzt werden:

\relative {
  f'1 \bar "|"
  f1 \bar "."
  g1 \bar "||"
  a1 \bar ".|
  b1 \bar ".".
  c1 \bar ".|.
  d1 \bar ".|
  e1
}

Zusätzlich gibt es noch punktierte und gestrichelte Taktstriche:

\relative {
  f'1 \bar ";"
  g1 \bar "!
  a1
}

und fünf unterschiedliche Wiederholungstaktstriche:

f1 \bar ".:" g \bar ":..:" a \bar ":.:" b \bar ":.:" c \bar ":.:" d

Zusätzlich kann eine Taktlinie mit einem einfachen Apostroph gesetzt werden:

f1 \bar ".:" g1 \bar ":..:"
Zusätzlich kann ein Taktstrich auch nur als kleines Komma gesetzt werden:

\f1 \bar """"

Derartige Apostrophe werden allerdings vor allem im gregorianischen Choral eingesetzt, und es wird empfohlen, anstatt dessen \textit{divisioMinima} zu benutzen, wie beschrieben im Abschnitt [Divisiones], Seite 423.

Für \textit{segno}-Zeichen innerhalb des Systems gibt es drei Taktstricharten, die sich in ihrem Verhalten an Zeilenumbrüchen unterscheiden:

\begin{verbatim}
\relative c'' {
    c4 c c c
    \bar "S"
    c4 c c c \break
    \bar "S"
    c4 c c c
    \bar "S-|"
    c4 c c c \break
    \bar "S-|"
    c4 c c c
    \bar "S-S"
    c4 c c c \break
    \bar "S-S"
    c1
}
\end{verbatim}
Auch wenn die Taktlinien, die Wiederholungen angeben, manuell eingefügt werden können, wird die Wiederholung dadurch nicht von LilyPond erkannt. Wiederholte Stellen werden besser notiert, indem man die Wiederholungs-Befehle einsetzt, die automatische die richtigen Taktlinien setzen. Das ist beschrieben in Abschnitt 1.4 [Wiederholungszeichen], Seite 136.

Zusätzlich kann noch "||:" verwendet werden, dass sich genauso wie "|:" verhält, außer bei Zeilenumbrüchen, wo ein doppelter Taktstrich am Ende der Zeile ausgegeben wird und ein öffnender Wiederholungsstrich am Anfang der nächsten Zeile.

Für Kombinationen von Wiederholungen mit dem segno-Zeichen gibt es sechs verschiedene Variationen:

```latex
\relative c' {  
c4 c c c  
\bar "|.S"  
c4 c c c \break  
\bar "|.S"  
c4 c c c  
\bar "|.S-S"  
c4 c c c \break  
\bar "|.S-S"  
c4 c c c  
\bar "S.|:-S"  
c4 c c c \break  
\bar "S.|:-S"  
c4 c c c  
\bar "S.|:"  
c4 c c c \break  
\bar "S.|:"  
c4 c c c \break  
\bar "S.|:-S"  
c4 c c c \break  
\bar "S.|:-S"
}
```
Darüber hinaus wählt der \texttt{inStaffSegno}-Befehl eines dieser Segno-Taktstriche aus, in Zusammenarbeit mit dem \texttt{repeat volta}-Befehl.

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

\begin{verbatim}
<<
\new StaffGroup <<
\new Staff \relative { e'4 d \bar "||" f4 e }
\new Staff \relative { \clef bass c'4 g e g }
>>
\new Staff \relative { \clef bass c'2 c2 }
>>
\end{verbatim}
Ausgewählte Schnipsel

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

Der automatisch erzeugte Taktstrich ist "|". Das kann jederzeit durch den Befehl \set Timing\.defaultBarType = Taktstrichart geändert werden.

Siehe auch

Notationsreferenz: Abschnitt 4.3.1 [Zeilenumbrüche], Seite 512, Abschnitt 1.4 [Wiederholungszeichen], Seite 136, [Systeme gruppieren], Seite 175.

Schnipsel: Abschnitt “Rhythms” in Schnipsel.


Taktzahlen

Taktzahlen werden standardmäßig zu Beginn eines jeden Systems ausgegeben, ausgenommen ist die erste Zeile. Die Zahl selber wird in der currentBarNumber-Eigenschaft gespeichert, die normalerweise für jeden Takt aktualisiert wird. Sie kann aber auch manuell gesetzt werden:

\begin{verbatim}
c1 c c c \break
\set Score\.currentBarNumber = #50
\end{verbatim}

c1 c c c

Taktnummern können in regelmäßigen Abstand ausgegeben werden, anstatt dass sie nur am Beginn des Systems erscheinen. Um das zu erreichen, muss die Standardeinstellung verändert werden, um zu erlauben, dass Taktnummern an anderen Stellen als dem Beginn von Systemen ausgegeben werden. Das wird mit der Eigenschaft break-visibility von BarNumber vorgenommen. Sie braucht drei Werte, die auf #t (wahr) oder #f (falsch) gestellt werden können, womit angegeben wird, ob die Taktnummer an der entsprechenden Stelle sichtbar ist. Die Reihenfolge der Werte ist: Ende der Zeile, Mitte der Zeile und Beginn der Zeile. Im folgenden Beispiel werden die Taktlinien überall ausgegeben:

\begin{verbatim}
\override Score\.BarNumber\.break-visibility = ##(#t #t #t)
\set Score\.currentBarNumber = #11
\end{verbatim}
Setzen der Taktnummer für den ersten Takt

Standardmäßig wird die erste Taktnummer einer Partitur nicht gesetzt, wenn sie weniger oder gleich 1 ist. Indem man barNumberVisibility auf all-bar-numbers-visible setzt, kann eine beliebige Taktnummer für den ersten und die folgenden Takte gesetzt werden. Eine leere Taktlinie muss jedoch vor der ersten Note eingefügt werden, damit das funktioniert.

\layout {
  indent = 0
  ragged-right = ##t
}
\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \bar ""
  c1 | d | e | f \break
  g1 | e | d | c
}

Setzen der Taktnummern in regelmäßigen Intervallen

Taktnummern können in regelmäßigen Intervallen gesetzt werden, indem man die Eigenschaft barNumberVisibility definiert. In diesem Beispiel werden die Taktnummern jeden zweiten Takt gesetzt, außer am Ende einer Zeile.

\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = #11
  \% Permit first bar number to be printed
  \bar ""
  \% Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
Alternative Taktnummerierung

Zwei alternative Methoden können eingestellt werden, die die Taktnummerierung beeinflussen, insbesondere bei Wiederholungen.

\set Score.alternativeNumberingStyle = #'numbers
\repeat volta 3 { c4 d e f | }
alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
}
c1 \break
\set Score.alternativeNumberingStyle = #'numbers-with-letters
\repeat volta 3 { c,4 d e f | }
alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
}
c1
\set Score.alternativeNumberingStyle = #'numbers-with-letters


Setzen von Taktnummern in Kästchen oder Kreisen

Taktnummern können auch in Boxen oder Kreisen gesetzt werden.

\relative c' {
  % Prevent bar numbers at the end of a line and permit them elsewhere
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)
  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = #2
  % Draw a box round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \repeat unfold 5 { c1 }
  % Draw a circle round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 4 { c1 } \bar "."
}

Taktnummern ausrichten

Taktnummern sind standardmäßig links an ihrem Ursprungsobjekt ausgerichtet. Das ist normalerweise die linke Ecke einer Linie oder, wenn die Nummern innerhalb einer Zeile gesetzt werden, auf der linken Seite eines Taktstrichs. Die Nummern können auch direkt über dem Taktstrich positioniert werden oder rechts vom Taktstrich gesetzt werden.

\relative c' {
  \set Score.currentBarNumber = #111
  \override Score.BarNumber.break-visibility = #all-visible
  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = #2
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c1
  % Center-align bar numbers
  \override Score.BarNumber.self-alignment-X = #CENTER
Kapitel 1: Musikalische Notation

Entfernung von Taktnummern in einer Partitur

Taktnummern können vollkommen aus den Noten entfernt werden, indem man den Bar_number_engraver aus dem Score-Kontext entfernt.

\layout {
\context {\Score \omit BarNumber
% or:
%\remove "Bar_number_engraver"
}
}
relative c'' {
c4 c c c \break
c4 c c c
}

Siehe auch
Schnipsel: Abschnitt “Rhythms” in Schnipsel.

Referenz der Interna: Abschnitt “BarNumber” in Referenz der Interna, Abschnitt “Bar_number_engraver” in Referenz der Interna.

Bekannte Probleme und Warnungen
Taktnummern können mit der oberen Ecke der Klammer zu Beginn des Systems zusammenstoßen. Um das zu verhindern, kann die padding-Eigenschaft von BarNumber verwendet werden, um die Zahl zu verschieben. Für mehr Information siehe Abschnitt “StaffGroup” in Referenz der Interna und Abschnitt “BarNumber” in Referenz der Interna.
Takt- und Taktzahlüberprüfung


\time 3/4 c2 e4 \| g2 |

Taktüberprüfungen können auch in Gesangstexten verwendet werden:

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

Besonders in mehrstimmiger komplizierter Musik können falschen Notenwerte die ganze Partitur durcheinander bringen. Es lohnt sich also, die Fehlersuche damit zu beginnen, nicht bestandene Taktüberprüfungen zu kontrollieren.

Wenn aufeinander folgende Taktüberprüfungen mit dem gleichen Abstand Fehler produzieren, wird eventuell nur die erste Warnung ausgegeben. Damit wird die Warnung auf den Ursprung des Fehlers fokussiert.

Es ist auch möglich, die Bedeutung des Symbols „|“ (Pipe) umzudefinieren, so dass hiermit eine andere Aktion als eine Taktüberprüfung erreicht wird. Das geschieht, indem man der Pipe (*\|* einen musikalischen Ausdruck zuweist. Im nächsten Beispiel wird *\|* dazu verwendet, eine doppelte Taktlinie auszugeben, wo immer man das Zeichen auch setzt. Gleichzeitig hört das Zeichen auf, als Taktüberprüfung zu funktionieren.

\| = \bar "\|"

{  
c'2 c' |
c'2 c' 
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 Originale 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.

Siehe auch

Schnipsel: Abschnitt “Rhythms” in Schnipsel.

Übungszeichen

Übungszeichen können mit dem \mark-Befehl ausgegeben werden:

\mark \default
\mark \default
Das Zeichen wird automatisch um einen Wert heraufgesetzt, wenn man \mark \default benutzt, aber man kann auch eine Ganzzahl als Argument einsetzen, wenn man das Zeichen manuell setzen will. Der Wert, der eingesetzt werden soll, wird in der Eigenschaft rehearsalMark gespeichert.

\relative c'' {
    \set Score.markFormatter = #format-mark-alphabet
    \set Score.markFormatter = #format-mark-box-alphabet
    \set Score.markFormatter = #format-mark-circle-alphabet
    \relative c'' {
        \set Score.markFormatter = #format-mark-numbers
    }
}


\relative c'' {
    \set Score.markFormatter = #format-mark-box-alphabet
    \set Score.markFormatter = #format-mark-circle-alphabet
    \relative c'' {
        \set Score.markFormatter = #format-mark-numbers
    }
}
Kapitel 1: Musikalische Notation


Die Funktionen format-mark-numbers, format-mark-box-numbers und format-mark-circle-numbers können eingesetzt werden, um Taktnummern anstelle der fortlaufenden Zahlen bzw. Buchstaben zu erhalten.

Andere Übungszeichenstile können auch manuell gesetzt werden:

\mark "A1"

Score.markFormatter hat keine Auswirkungen auf solcherart definierte Zeichen. Man kann aber auch mit \markup Textbeschriftungsobjekte zu dem selbstdefinierten Zeichen hinzufügen:

\mark \markup{ \box A1 }

Musikbuchstaben (wie etwa das Segno-Zeichen) können mit dem Befehl \musicglyph als ein \mark-Zeichen definiert werden:

\relative c' {
  c1 \mark \markup{ \musicglyph "scripts.segno" }
  c1 \mark \markup{ \musicglyph "scripts.coda" }
  c1 \mark \markup{ \musicglyph "scripts.ufermata" }
}

Die Datei scm/translation-functions.scm enthält die Definitionen von format-mark-numbers und format-mark-letters. Sie können als Anregung für andere Formatierungsfunktionen genommen werden.
1.2.6 Besondere rhythmische Fragen

Verzierungen

Verzierungen, mit dem Befehl `\grace` notiert, sind ausgeschriebene Ornamente. Sie werden in einer kleineren Schriftgröße gesetzt und nehmen keine logische Zeit im Takt ein.

\[\text{relative} \left\{ \begin{array}{l}
  \text{c''4 \grace b16 a4} \\
  \text{\grace \{ b16 c16 \} a2}
\end{array} \right.\]


\[\text{relative} \left\{ \begin{array}{l}
  \text{acciaccatura d''8 c4} \\
  \text{appoggiatura e8 d4} \\
  \text{acciaccatura \{ g16 f \} e2} \\
  \text{\slashedGrace a,8 g4} \\
  \text{\slashedGrace b16 a4(} \\
  \text{\slashedGrace b8 a2)}
\end{array} \right.\]

Die Position von Verzierungen ist zwischen Notensystemen synchronisiert. Im nächsten Beispiel stehen im einen System zwei 16-Noten für jede 8-Note des zweiten Systems:

<< \new Staff \{ e2 \grace \{ c16 d e f \} e2 \} \\
\new Staff \{ c2 \grace \{ g8 b \} c2 \} >>
Wenn Sie eine Note mit einer Verzierung abschließen wollen, müssen Sie den `afterGrace`-Befehl benutzen. Er benötigt zwei Argumente: die Hauptnote und die Verzierung, die nach der Hauptnote folgen soll:

\begin{verbatim}
relative { c''1 afterGrace d1 { c16[ d ] } c1 }
\end{verbatim}


<<
\begin{verbatim}
new Staff \relative {
    c''1 afterGrace d1 { c16[ d ] } c1
}
new Staff \relative {
    #(define afterGraceFraction (cons 15 16))
    c''1 afterGrace d1 { c16[ d ] } c1
}
new Staff \relative {
    #(define afterGraceFraction (cons 1 2))
    c''1 afterGrace d1 { c16[ d ] } c1
}
>>

Der Abstand zwischen der Hauptnote und der Verzierung kann auch mit unsichtbaren Noten beeinflusst werden. Im nächsten Beispiel wird die Verzierung mit einem Abstand von 7/8 zur Hauptnote gesetzt.

\begin{verbatim}
new Voice {
    \relative {
        \afterGrace d1 ^ { \texttrill } \{ s2 s4. \grace { c16 d } \} \afterGrace c1
    }
}
\end{verbatim}
Ein \texttt{\textbackslash 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. Die Veränderungen müssen auch innerhalb des Verzierungsausdrucks rückgängig gemacht werden. In diesem Fall wird die Richtung der Hälse geändert und dann wieder der Standard eingestellt:

\new Voice \relative {
  \acciaccatura {
    \stemDown
    f''\textbackslash 16->
    \stemNeutral
  }\textbackslash 16
  g4 e c2
}

Ausgewählte Schnipsel

\emph{Using grace note slashes with normal heads}

The slash through the stem found in acciaccaturas can be applied in other situations.

\relative c' {
  \override Flag.stroke-style = "grace"
  c8( d2) e8( f4)
}

Veränderung des Layouts von Verzierungen innerhalb der Noten

Das Layout von Verzierungsausdrücken kann in der Musik verändert werden mit den Funktionen \texttt{add-grace-property} und \texttt{remove-grace-property}. Das folgende Beispiel definiert die Richtung von Hälsern (Stem) für diese Verzierung, sodass die Hälse nicht immer nach unten zeigen, und ändert den Standardnotenkopf in ein Kreuz.

\relative c' {
  \new Staff {
    $(\texttt{\textbackslash remove-grace-property 'Voice 'Stem 'direction})$
    $(\texttt{\textbackslash add-grace-property 'Voice 'NoteHead 'style 'cross})$
    \new Voice {
      \acciaccatura \textbackslash 16 g4
      \grace \textbackslash 16 e f4
      \appoggiatura \textbackslash 16 g a e2
    }
  }
}

\new Voice \relative {
  \acciaccatura \textbackslash 16 g4
  \grace \textbackslash 16 e f4
  \appoggiatura \textbackslash 16 g a e2
}
Globale Umdefinition von Verzierungsnoten


```
startAcciaccaturaMusic = {
  <>
    \override Flag.stroke-style = #"grace"
    \slurDashed
}

stopAcciaccaturaMusic = {
  \revert Flag.stroke-style
  \slurSolid
  <>
}

\relative c' { 
  \acciaccatura d8 c1
}
```

Positionierung von Verzierungen mit verschiebbarem Platz

Wenn man die Eigenschaft `strict-grace-spacing` aktiviert, werden die Verzierungsnoten "fließend" gemacht, d.h. sie sind von den normalen Noten los gekoppelt: Zuerst werden die normalen Noten platziert, dann erst die Verzierungen links von der Hauptnote gesetzt.

```
\relative c' {
  <<
    \override Score.SpacingSpanner.strict-grace-spacing = ##t
    \new Staff \new Voice {
      \afterGrace c4 { c16[ c8 c16 ] }
      c8[ \grace { b16 d } c8]
      c4 r
    }
    \new Staff {
      c16 c c c c c c c c4 r
    }
  >>
}
```

Siehe auch


Notationsreferenz: [Tondauern skalieren], Seite 48, [Manuelle Balken], Seite 88.

Installierte Dateien: ly/grace-init.ly

Schnipsel: Abschnitt “Rhythms” in Schnipsel.


Bekannte Probleme und Warnungen

Ein Vorschlag (acciaccatura) mit mehreren Noten und Balken wird ohne den Schrägestrich gesetzt und sieht einem Vorhalt (appoggiatura) sehr ähnlich.

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:

```latex
<<
  \new Staff \relative { e''4 \bar ".|:" \grace c16 d2. }
  \new Staff \relative { c''4 \bar ".|:" d2. }
>>
```

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

```latex
<<
  \new Staff { e4 \bar ".|:" \grace c16 d2. }
  \new Staff { c4 \bar ".|:" \grace s16 d2. }
>>
```

gesetzt werden.


```latex
accMusic = {
  \acciaccatura { f8 } e8 r8 \acciaccatura { f8 } e8 r4
}
```
Verzierungsabschnitte sollten nur innerhalb von sequentiellen musikalischen Ausdrücken benützt werden. Wenn sie ineinandergeschachtelt werden, kann es zu Fehlermeldungen oder Abstürzen kommen.

Jede Verzierungsnote in der MIDI-Ausgabe hat ein Viertel der Dauer ihrer wirklichen Dauer. Wenn die addierte Dauer der Verzierungsnoten länger als die Dauer der vorhergehenden Note dauert, wird der Fehler „Going back in MIDI time“ ausgegeben. Man muss die Verzierungsnoten entweder kürzer machen, etwa:

\acciaccatura \{ c'8[ d' e' f' g'] \}

wird zu:

\acciaccatura \{ c'16[ d' e' f' g'] \}

oder die Dauern explizit ändern:

\acciaccatura \{ \scaleDurations 1/2 \{ c'8[ d' e' f' g'] \} \}

Siehe [Tondauern skalieren], Seite 48.

An Kadenzen ausrichten
In Orchesterpartituren stellen Kadenzen ein besonderes Problem dar: Wenn in der Partitur ein Instrument eine Kadenz spielt, die notiert wird, müssen die anderen Stimmen genau die entsprechende Anzahl Noten überspringen, damit sie nicht zu früh oder zu spät einsetzen.

Eine Lösung ist es, die Funktionen \mmrest-of-length oder \skip-of-length zu benutzen. Diese Scheme-Funktionen brauchen einen definierten Notenabschnitt (eine Variable) als Argument und produzieren entweder Ganztaktpausen oder leere Takte, die genauso lang sind wie der Notenabschnitt.

\MyCadenza = \relative {\c4 \d8 e f g g4 \f2 g4 g}
Kapitel 1: Musikalische Notation

\new GrandStaff <<
\new Staff {
   \MyCadenza c'1
   \MyCadenza c'1
}
\new Staff {
   #(mmrest-of-length MyCadenza)
   c'1
   #(skip-of-length MyCadenza)
   c'1
}
>>

Siehe auch
Glossar: Abschnitt “cadenza” in Glossar.
Schnipsel: Abschnitt “Rhythms” in Schnipsel.

Verwaltung der Zeiteinheiten

Die folgenden Eigenschaften von Timing werden eingesetzt, um die Zeit in Partituren zu verwalten.

currentBarNumber (aktuelle Taktnummer)
Die gerade aktuelle Taktzahl. Für ein Beispiel, das die Benutzung dieser Eigenschaft zeigt, siehe [Taktzahlen], Seite 97.

measureLength (Taktlänge)

measurePosition (Taktposition)
Der Schlag im Takt zum aktuellen Moment. Dieser Wert wird zurückgesetzt, indem measureLength (die Taktlänge) abgezogen wird, wenn der Wert von measureLength erreicht oder überschritten wird. Wenn das passiert, wird der Zähler currentBarNumber (aktuelle Taktnummer) erhöht.

timing (Zeitberechnung)
Wenn auf wahr gesetzt, werden die oben genannten Variablen zu jedem Zeitpunkt aktualisiert. Wenn auf falsch gesetzt, bleibt der Engraver unendlich lange im aktuellen Takt.

\set Score.measureLength = #(ly:make-moment 5/4)
c1 c4
c4 c4
\set Score.measurePosition = #(ly:make-moment 5/8)
b4 b4 b8
c4 c1


Siehe auch

Notationsreferenz: [Taktzahlen], Seite 97, [Musik ohne Metrum], Seite 69.
Schnipsel: Abschnitt “Rhythms” in Schnipsel.
Referenz der Interna: Abschnitt “Timing translator” in Referenz der Interna, Abschnitt “Score” in Referenz der Interna

1.3 Ausdrucksbezeichnungen
Dieser Abschnitt zeigt verschiedene Ausdrucksbezeichnungen, die zur Partitur hinzugefügt werden können.

1.3.1 Ausdrucksbezeichnungen an Noten angehängt

Dieser Abschnitt erklärt, wie man Ausdrucksbezeichnungen erstellt, die an Noten gebunden sind: Artikulationszeichen, Ornamente und Dynamikzeichen. Es werden auch Methoden gezeigt, eigene Ausdrucksbezeichnungen zu erstellen.

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. Hierzu wird folgende Syntax benutzt:

\relative {  
c''4\staccato c\mordent b2\turn  
c1\fermata  
}

Einige dieser Artikulationszeichen haben eine Abkürzung, damit es einfacher ist, sie zu schreiben. Die Abkürzung wird an die Notenbezeichnung gehängt, wobei ihre Syntax aus einem Minuszeichen - besteht, gefolgt von dem Symbol, das dem Artikulationszeichen zugeordnet ist. Es gibt diese Abkürzungen für marcato, stopped (gedämpft), tenuto, staccatissimo, accent, staccato, und portato. Die ihnen entsprechenden Symbole werden also folgendermaßen notiert:

\relative {  
c''4^- c+- c-- c-!  
c4-> c-. c2_-  
}

Die Regeln für die standardmäßige Platzierung von Artikulationszeichen werden in der Datei scm/script.scm definiert. Artikulationszeichen und Ornamente können manuell über oder unter dem System gesetzt werden, siehe Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

Artikulationszeichen sind Script-Objekte. Ihre Eigenschaften werden ausführlich in Abschnitt “Script” in Referenz der Interna beschrieben.

Artikulationen können neben Noten auch an Pausen gehängt werden, aber sie können nicht an Mehrtaktpausen gehängt werden. Ein besonderer Befehl, fermata Markup, wurde definiert,
damit man eine Fermate an eine Mehraktapause anfügen kann (und nur hieran). Damit wird ein MultiMeasureRestText-Objekt erstellt.

\override Script.color = #red
\override MultiMeasureRestText.color = #blue
\relative c'' { c1-+ } \fermata
R1\fermata

Zusätzlich zu den Artikulationszeichen können auch Text und Beschriftung an Noten angehängt werden. Siehe auch [Textarten], Seite 216.

Zu weiterer Information über die Reihenfolge von Scripten und TextScripten, die an Noten angehängt werden, siehe Abschnitt “Positionierung von Objekten” in Handbuch zum Lernen.

**Ausgewählte Schnipsel**

*Die Standardwerte der Abkürzungen von Artikulationen verändern*


\relative c'' { c1-+ }

\relative c'' { c1+- }

Die vertikale Anordnung von Beschriftungen kontrollieren

Die vertikale Anordnung von Beschriftungen wird mit der 'script-priority-Eigenschaft kontrolliert. Um so kleiner die Zahl, um so näher wird die Beschriftung in Bezug auf die Note gesetzt. In diesem Beispiel hat das TextScript-Objekt (das Kreuz) zuerst die niedrigste Priorität, wird also auch am niedrigsten in dem ersten Beispiel gesetzt. Im zweiten Fall hat der Praller (das Script) die niedrigste Priorität, darum wird er am nächsten zum System gesetzt. Wenn zwei Objekte die gleiche Priorität haben, wird ihre Reihenfolge anhand ihres Auftretens in der Quelldatei entschieden.

\relative c'' {
    \once \override TextScript.script-priority = #-100
    a2^\prall\markup { \sharp }
\}

\once \override Script.script-priority = #-100
Einen Doppelschlag mit Vorhalt erstellen

Einen Doppelschlag mit Vorhalt zu erstellen, wobei die untere Note das Vorzeichen benutzt, erfordert einige Einstellungänderungen. Die outside-staff-priority-Eigenschaft muss auf falsch (#f) gesetzt werden, weil sie sonst über die Eigenschaft avoid-slur property dominieren würde. Der Wert von halign wird benutzt, um den Doppelschlag horizontal zu positionieren.

\relative c'' {{c2*2/3 (s2*1/3\turn d4) r} << \{c4.(d8)\} \{s4 s\turn\} >> \transpose c d \relative c'' << \{c4.(d8)\} \{s4 \once \set suggestAccidentals = ##t \once \override AccidentalSuggestion.outside-staff-priority = ##f \once \override AccidentalSuggestion.avoid-slur = #'inside \once \override AccidentalSuggestion.font-size = -3 \once \override AccidentalSuggestion.script-priority = -1 \single \hideNotes b8-\turn \noBeam s8 \}}

Siehe auch


Notationsreferenz: [Textarten], Seite 216, Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581, Abschnitt A.13 [Liste der Artikulationszeichen], Seite 705, [Triller], Seite 133.

Installierte Dateien: scm/script.scm


Dynamik

Absolute Dynamikbezeichnung wird mit Befehlen nach den Noten angezeigt, etwa c4\ff. Die vordefinierten Befehle lauten: \pppp, \pppp, \ppp, \pp, \mp, \mf, \f, \ff, \fff, \ffff, \fz, \sz, \sp, \spz, \spp, \sppz, \sff, \sf, \sff, \sp, \spf, \sms, \sfz, \smsz, \spp, \sppf, \spz. Die Dynamikzeichen können manuell unter- oder oberhalb des Systems platziert werden, siehe Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

\relative c'' {
  c2\ppp c\mp
  c2\sz c^\mf
  c2\sms c^\ff
}

Eine Crescendo-Klammer wird mit dem Befehl \< begonnen und mit \!, einem absoluten Dynamikbefehl oder einer weiteren Crescendo- oder Decrescendo-Klammer beendet. Ein Decrescendo beginnt mit \> und wird auch beendet mit \!, einem absoluten Dynamikbefehl oder einem weiteren Crescendo oder Decrescendo. \cr und \decr können anstelle von \< und \> benutzt werden. Die Befehle ergeben standardmäßig Crescendo-Klammern.

\relative c'' {
  c2\< c!
  d2\< d\f
  e2\< e\>
  f2\> f!
  e2\> e\mp
  d2\> d\>
  c1!
}

Eine Crescendo-Klammer, die mit \! beendet wird, endet an der rechten Seite der Note, welcher \! zugeordnet ist. In dem Fall, dass es durch den Beginn eines anderen crescendo- oder decrescendo-Zeichens beendet wird, endet es in der Mitte der Note, welche das nächste \< oder \> angehängt hat. Die nächste Klammer beginnt dann am rechten Rand der selben Note anstatt dem normalerweise linken Rand, wenn die vorherige Klammer mit \! beendet worden wäre.

\relative {
  c'\<1| c4 a c\< a | c4 a c\! a\< | c4 a c a!}

\relative {
  c'\<1| c4 a c\< a | c4 a c\! a\< | c4 a c a!}
Leere Pausenzeichen werden benötigt, um mehrere Zeichen für eine Note zu notieren. Das ist insbesondere nützlich, wenn man crescendo und decrescendo zu der selben Note hinzufügen will:

\relative {
  c'\!4\!< c\! \ d \> e\!
  << f1 \{ s4 \ s4\!< \ s4\> \ s4\! \} >>
}

Der \espressivo-Befehl kann eingesetzt werden, um crescendo und decrescendo für die selbe Note anzuzeigen. Dieser Befehl ist jedoch als Artikulation, nicht als Dynamikzeichen implementiert.

\relative {
  c''2 \ b \ a  \
g1\espressivo
}

Mit Text gesetzte Crescendo-Bezeichnungen beginnen mit \cresc. Mit Text gesetzte Decrescendo-Bezeichnungen beginnen mit \decresc oder \dim. Fortsetzungslinien werden gesetzt, wenn sie benötigt werden.

\relative {
  g'8\cresc a b c b c d e\!mf |
  f8\decresc e d c e\> d c b |
  a1\dim - |
  a2. r4\! |
}

Als Text gesetzte Dynamik-Bezeichnungen können auch die Crescendo-Klammern ersetzen:

\relative c' \ {
  \crescTextCresc
  c4\!< d e f\! |
  \dimTextDecresc
  g4\> e d c\! |
  \dimTextDecr
  e4\> d c b\! |
  \dimTextDim
  d4\> c b a\! |
  \crescHairpin
  \dimHairpin
}
Um neue absolute Dynamikzeichen oder Text, der mit ihnen angeordnet wird, zu erstellen, siehe [Neue Lautstärkezeichen], Seite 121.


Es gibt einen besonderen Dynamics-Kontext, um Crescendi und Decrescendi auf einer eigenen Zeile zu notieren. Mit leeren Pausen (s) werden die Dauern gesetzt. (Auch Noten in einem Dynamics-Kontext nehmen eine Dauer ein, werden aber nicht gesetzt.) Der Dynamics-Kontext ist sehr nützlich, um andere Elemente wie Textbeschriftung, Text-Strecker und Klavierpedalbezeichnungen aufzunehmen.

\[
\begin{align*}
\text{c'} &\ 2 \ d4 \ e | \\
\text{c} &\ 4 \ e, \ 2 | \\
\text{g'} &\ 4 \ a \ g \ a | \\
\text{c} &\ 1 | \\
\end{align*}
\]

\[
\begin{align*}
\text{s} &\ 1< | \\
\text{s} &\ 1\f | \\
\text{s} &\ 2\dim \ s2-"rit." | \\
\text{s} &\ 1\p | \\
\end{align*}
\]

\[
\begin{align*}
f &\ \text{dim. rit.} \ - \ p
\end{align*}
\]

Vordefinierte Befehle

\\textbackslash dynamicUp, \textbackslash dynamicDown, \textbackslash dynamicNeutral, \textbackslash crescTextCresc, \textbackslash dimTextDim, \textbackslash dimTextDecr, \textbackslash dimTextDecresc, \textbackslash crescHairpin, \textbackslash dimHairpin.

Ausgewählte Schnipsel

Das Verhalten von Crescendo-Klammern an Taktlinien beeinflussen

Wenn die Note, an welcher eine Crescendo-Klammer endet, die erste Note eines Taktes ist, wird die Klammer an der vorhergehenden Taktlinie beendet. Dieses Verhalten kann auch mit der Eigenschaft 'to-barline geändert werden:

\[
\begin{align*}
\text{relative c'} &\ {'} \\
\text{e} &\ 4< \ e2. \\
\text{e} &\ 1! \\
\text{\override \Hairpin.to-barline = ##f}
\end{align*}
\]
Die Mindestlänge von Crescendo-Klammern bestimmen
Wenn Crescendo-Klammern zu kurz sind, können sie verlängert werden, indem die `minimum-length`-Eigenschaft des `Hairpin`-Objektes verändert wird.

```latex
\relative c'' {
  \override Hairpin.minimum-length = #5
  << f1 { s4 s\< s\> s\! } >>
}
```

Crescendo Klammern al niente schreiben
Crescendo-Klammern können mit einem kleinen Kreis vor der Spitze notiert werden (al niente = bis zum Nichts), indem die `circled-tip`-Eigenschaft des `Hairpin`-Objekts auf `#t` gesetzt wird.

```latex
\relative c'' {
  \override Hairpin.circled-tip = ##t
  c4\< c\! c\> c\< c2\!
}
```

Vertikale Ausrichtung von Dynamik und Textbeschriftung beeinflussen

Die gleiche Idee wird benutzt, um Textbeschriftungen an ihrer Grundlinie auszurichten.

```latex
music = \relative c' {
  a'2\p b\f\> g, b\p
c2`\markup { \huge gorgeous } c`\markup { \huge fantastic }
}
```
Crescendo-Linien von Dynamik-Texten unterdrücken

Dynamik-Texte (wie cresc. und dim.) werden mit einer gestrichelten Linie gesetzt, die ihre Dauer anzeigt. Diese Linie kann auf folgende Weise unterdrückt werden:

```music
\relative c'' {
  \override DynamicTextSpanner.style = #'none
  \crescTextCresc
  c1< | d | b | c!
}
```

Text und Strecker-Stile für Dynamik-Texte ändern


```music
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2< a
  a2 a
  a2 a\mf
}
```
Siehe auch


Notationsreferenz: Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581, [Neue Lautstärkezeichen], Seite 121, Abschnitt 3.5.3 [Was geht in die MIDI-Ausgabe], Seite 490, Abschnitt 3.5.5 [MIDI-Lautstärke kontrollieren], Seite 491.


Neue Lautstärkezeichen

Die einfachste Art, eigene Dynamikbezeichnungen zu erstellen, ist die Benutzung von \markup-(Textbeschriftungs)-Objekten.

\markup{\relative{\moltoF \c'1 \sfzp c1}}

Mit einer Textbeschriftung können editorische Dynamikzeichen (in runden oder eckigen Klammern) erstellt werden. Die Syntax für den Textbeschriftungsmodus wird erklärt in Abschnitt 1.8.2 [Text formatieren], Seite 223.

\markup{\center-align \concat \bracket \dynamic f}
Allgemein gesagt kann `make-dynamic-script` jegliches Textbeschriftungsobjekt als Argument haben. Die Schriftart für Dynamikzeichen enthält nur die Buchstaben f,m,p,r,s sowie z; ein Dynamikzeichen, das anderen Text oder Satzzeichen enthalten soll, benötigt Textbeschriftungsbefehle, die die normale Schriftart einschalten, etwa `\normal-text`. Die Funktion `make-dynamic-script` sollte anstelle einer normalen Textbeschriftung vor allem deshalb benutzt werden, weil auf diese Weise die vertikale Ausrichtung von den Textbeschriftungen (engl. markup) und den spitzen Klammern an der selben Linie gewährleistet wird.

```
roundF = \markup { \center-align \concat {
    \normal-text { \bold { \italic ( } }
    \dynamic f
    \normal-text { \bold { \italic ) } } }
}
boxF = \markup { \bracket { \dynamic f } }
mfEspress = \markup { \center-align \line {
    \hspace #3.7 mf \normal-text \italic espress. } }
roundFdynamic = #(make-dynamic-script roundF)
boxFdynamic = #(make-dynamic-script boxF)
mfEspressDynamic = #(make-dynamic-script mfEspress)
```

\relative {
    c'4\roundFdynamic\< d e f
g,1\boxFdynamic\>
g1
g'1\mfEspressDynamic
g1
}

Anstelle dessen kann auch die Scheme-Form des Beschriftungs-Modus verwendet werden. Seine Syntax ist erklärt in Abschnitt "Beschriftungskonstruktionen in Scheme" in `Extending`.

```
moltoF = #(make-dynamic-script
    (markup #:normal-text "molto"
        #:dynamic "f")
)
```

\relative {
    \<d' e>16 \<d e>
    \<d e>2..\moltoF
}

Die Auswahl von Schriftarten in Textbeschrifungen ist erklärt in [Überblick über die wichtigsten Textbeschriftungsbefehle], Seite 225.
Siehe auch

Notationsreferenz: Abschnitt 1.8.2 [Text formatieren], Seite 223, [Überblick über die wichtigsten Textbeschriftungsbefehle], Seite 225, Abschnitt 3.5.3 [Was geht in die MIDI-Ausgabe], Seite 490, Abschnitt 3.5.5 [MIDI-Lautstärke kontrollieren], Seite 491.


1.3.2 Ausdrucksbezeichnungen als Bögen

Dieser Abschnitt erklärt, wie man verschiedene gebogene Ausdrucksbezeichnungen erstellt: Legato- und Phrasierungsbögen, Atemzeichen und Glissandos zu unbestimmten Tonhöhen.

Legatobögen

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

\relative {
   \slurDashed e'4( g a) a8 b( \\
   a4 g2 f4) \\
   \slurDotted <c e>2( <b d>2)
}

Legatobögen können manuell ober- oder unterhalb des Notensystems besetzt werden, siehe Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

Gleichzeitige, überlappende Legatobögen sind nicht erlaubt, aber ein Phrasierungsbogen kann einen Legatobogen überlappen. Damit können zwei Bögen gleichzeitig ausgegeben werden. Siehe auch [Phrasierungsbögen], Seite 126.

Legatobögen können durchgehend, gepunktet oder gestrichelt dargestellt werden. Standard ist der durchgehende Bogen:

\relative {
   \slurDashed c'4( e g2) \\
   \slurDotted g4( e c2) \\
   \slurSolid c4( e g2) \\
   \slurSolid g4( e c2)
}

Legatobögen können manuell ober- oder unterhalb des Notensystems besetzt werden, siehe Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.
Bögen können auch halb gestrichelt (die erste Hälfte gestrichelt, die zweite Hälfte durchgehend) erstellt werden, oder als halb durchgehend (die erste Hälfte durchgehend, die zweite Hälfte gestrichelt):

```
\relative {
    c'4( e g2)
    \slurHalfDashed
    g4( e c2)
    \slurHalfSolid
    c4( e g2)
    \slurSolid
    g4( e c2)
}
```

Eigene Muster für die Strichelung können definiert werden:

```
\relative {
    c'4( e g2)
    \slurDashPattern #0.7 #0.75
    g4( e c2)
    \slurDashPattern #0.5 #2.0
    c4( e g2)
    \slurSolid
    g4( e c2)
}
```

Vordefinierte Befehle

```
\slurUp, \slurDown, \slurNeutral, \slurDashed, \slurDotted, \slurHalfDashed, \slurHalfSolid, \slurDashPattern, \slurSolid.
```

Ausgewählte Schnipsel

*Doppelte Bögen für Legato-Akkorde benutzen*

Einige Komponisten schreiben doppelte Bögen, wenn Legato-Akkorde notiert werden. Das kann mit der Eigenschaft doubleSlurs erreicht werden.

```
\relative c' {
    \set doubleSlurs = ##t
    <c e>4( <d f> <c e> <d f>)
}
```

\[\text{\includegraphics{example.png}}\]
Textbeschriftung innerhalb von Bögen positionieren

Textbeschriftung kann innerhalb von Bögen gesetzt werden, wenn die outside-staff-priority-Eigenschaft auf falsch gesetzt wird.

\relative c'' {  
  \override TextScript.avoid-slur = #'inside  
  \override TextScript.outside-staff-priority = ##f  
  c2(\markup { \halign #-10 \natural } d4.) c8  
}\noteheads.s2\flags.d3\noteheads.s2\noteheads.s1\accidentals.natural\clefs.G\timesig.C44\dots.dot

Legatobögen mit kompliziertem Strichmuster definieren


Der Bogen wird nach dem Bezierparameter t definiert, welcher von 0 am linken Ende des Bogens zu 1 am rechten Ende des Bogens reicht. dash-element ist eine Liste (start-t stop-t dash-Unterbrechung dash-Abschnitt). Die Region des Bogens von start-t bis stop-t hat eine Unterbrechung von dash-Unterbrechung von jedem dash-Abschnitt-Schwarzbogen. dash-Abschnitt ist in Notenlinienzwischenräumen definiert. dash-Abschnitt ist auf 1 für einen durchgehenden Bogen gesetzt.

\relative c' {  
  \once \override  
  Slur.dash-definition = #'((0 0.3 0.1 0.75)  
    (0.3 0.6 1 1)  
    (0.65 1.0 0.4 0.75))  
  c4( d e f)  
  \once \override  
  Slur.dash-definition = #'((0 0.25 1 1)  
    (0.3 0.7 0.4 0.75)  
    (0.75 1.0 1 1))  
  c4( d e f)  
}\noteheads.s2\noteheads.s2\noteheads.s2\noteheads.s2\timesig.C44\clefs.G\timesig.C44\notes.s2\timesig.C44\flags.d3\timesig.C44\flags.d3\timesig.C44\flags.d3\timesig.C44\flags.d3\timesig.C44\flags.d3\timesig.C44\flags.d3\timesig.C44

Siehe auch

Glossar: Abschnitt “slur” in Glossar.
Notationsreferenz: Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581, [Phrasierungsbögern], Seite 126.
Referenz der Interna: Abschnitt “Slur” in Referenz der Interna.
Phrasierungsbögen

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

\[ \text{relative} \{
\text{c'}4\text{( e g2)\ phrasingSlurDashed}
g4\text{( e c2)\ phrasingSlurDotted}
c4\text{( e g2)\ phrasingSlurSolid}
g4\text{( e c2)}
\} \]

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. Phrasierungsbögen können manuell oberhalb oder unterhalb des Notensystems gesetzt werden, siehe Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

Simultane oder überlappende Phrasierungsbögen sind nicht erlaubt.

Phrasierungsbögen können durchgehend, gepunktet oder gestrichelt dargestellt werden. Standard ist der durchgehende Bogen:

\[ \text{relative} \{
\text{c'}4\text{( e g2)}
\phantom{2C.\ phrasingSlurDashed}
g4\text{( e c2)\ phrasingSlurDotted}
c4\text{( e g2)\ phrasingSlurSolid}
g4\text{( e c2)}
\} \]

Phrasierungsbögen können auch als halbgestrichelt dargestellt werden (die erste Hälfte gestrichelt, die zweite Hälfte durchgehend, oder halb durchgehend (die erste Hälfte durchgehend, die zweite gestrichelt):

\[ \text{relative} \{
\text{c'}4\text{( e g2)}
\phantom{2C.\ phrasingSlurHalfDashed}
g4\text{( e c2)\ phrasingSlurHalfSolid}
c4\text{( e g2)\ phrasingSlurSolid}
g4\text{( e c2)}
\} \]
Eigene Strichelmuster für Phrasierungsbögen können definiert werden:
\begin{verbatim}
\relative {
  c'4\( e g2\)
  \phrasingSlurDashPattern #0.7 #0.75
g4\( e c2\)
  \phrasingSlurDashPattern #0.5 #2.0
c4\( e g2\)
  \phrasingSlurSolid
g4\( e c2\)
}
\end{verbatim}

Strichelmusterdefinitionen für Phrasierungsbögen haben die gleiche Struktur wie die Definitionen für Legatobögen. Zu mehr Information über komplizierte Strichelmuster, siehe die Schnipsel im Abschnitt [Legatobögen], Seite 123.

Vordefinierte Befehle
\begin{verbatim}
\phrasingSlurUp, \phrasingSlurDown, \phrasingSlurNeutral, \phrasingSlurDashed, \phrasingSlurDotted, \phrasingSlurHalfDashed, \phrasingSlurHalfSolid, \phrasingSlurDashPattern, \phrasingSlurSolid.
\end{verbatim}

Siehe auch
Notationsreferenz: Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.
Referenz der Interna: Abschnitt “PhrasingSlur” in Referenz der Interna.

Atemzeichen
Atemzeichen werden mit dem Befehl \breathe eingegeben.
\begin{verbatim}
{ c''2. \breathe d''4 }
\end{verbatim}

Ein Atemzeichen bezeichnet gleichzeitig das Ende eines automatischen Balkens. Um das Verhalten zu verändern siehe [Manuelle Balken], Seite 88.
\begin{verbatim}
\relative { c''8 \breathe d e f g2 }
\end{verbatim}

Musikalische Zeichen für Atemzeichen in Alter Notation, auch Divisiones genannt, sind unterstützt. Für Einzelheiten siehe [Divisiones], Seite 423.
Ausgewählte Schnipsel

Das Atemzeichen-Symbol verändern

Das Schriftzeichen für das Atemzeichen kann verändert werden, indem die Text-Eigenschaft des `BreathingSign`-Layoutobjekts mit einer beliebigen Textbeschriftung definiert wird.

```latex
\relative c'' {
\override BreathingSign.text = \markup { \musicglyph "scripts.rvarcomma" }
\breathe
d2
}
```

Eine Zäsur einfügen

Zäsurzeichen können erstellt werden, indem die `text`-Eigenschaft des `BreathingSign`-Objektes verändert wird. Ein gekrümmtes Zäsurzeichen ist auch möglich.

```latex
\relative c'' {
\override BreathingSign.text = \markup { \musicglyph "scripts.caesura.straight" }
c8 e4. \breathe g8. e16 c4
\override BreathingSign.text = \markup { \musicglyph "scripts.caesura.curved" }
g8 e'4. \breathe g8. e16 c4
}
```

Siehe auch

Glossar: Abschnitt “caesura” in Glossar.
Notationsreferenz: [Divisiones], Seite 423.

Glissando zu unbestimmter Tonhöhe

Gleiten nach oben und unten kann mit dem Befehl `\bendAfter` notiert werden. Die Richtung des Glissandos wird mit einem Plus oder Minus (nach oben bzw. nach unten) angezeigt. Die Zahl zeigt die Intervallgröße an, über die sich das Glissando nach der Note erstreckt.

```latex
\relative c'' {
\bendAfter #+4
}
```
1.3.3 Ausdrucksbezeichnungen als Linien

Dieser Abschnitt zeigt, wie man verschiedene Ausdrucksbezeichnungen erstellt, die sich linear erstrecken: Glissando, Arpeggio und Triller.

Glissando

Ein **Glissando** wird mit dem Befehl \texttt{\glissando} auf eine Note folgend notiert:

\begin{verbatim}
g2\glissando g' 
c2\glissando c,
\end{verbatim}

Verschiedene Glissando-Stile sind möglich. Für Einzelheiten siehe Abschnitt 5.4.7 [Linienstile], Seite 594.
Ausgewählte Schnipsel

Glissando kann Grobs überspringen

NoteColumn-Grobs können bei Glissandos übersprungen werden.

\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}

Moderne Glissandi

Ein modernes Glissando ohne eine Endnote kann gesetzt werden, indem eine Kadenz eingesetzt wird und die Endnote unsichtbar gemacht wird.

\relative c'' {
  \time 3/4
  \override Glissando.style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
}

Siehe auch

Glossar: Abschnitt “glissando” in Glossar.
Notationsreferenz: Abschnitt 5.4.7 [Linienstile], Seite 594.
Referenz der Interna: Abschnitt “Glissando” in Referenz der Interna.

Bekannte Probleme und Warnungen

Printing text over the line (such as gliss.) is not supported.

Arpeggio

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

\relative { <c' e g c>1\arpeggio }
Kapitel 1: Musikalische Notation

Unterschiedliche Arpeggio-Typen können benutzt werden. \texttt{\textbackslash arpeggioNormal} stellt wieder das normale Verhalten her:

\begin{verbatim}
\relative {  
  <c' e g c>2\arpeggio

  \arpeggioArrowUp
  <c e g c>2\arpeggio

  \arpeggioArrowDown
  <c e g c>2\arpeggio

  \arpeggioNormal
  <c e g c>2\arpeggio
}
\end{verbatim}

Besondere Arpeggios mit Klammern können erstellt werden:

\begin{verbatim}
\relative {  
  <c' e g c>2

  \arpeggioBracket
  <c e g c>2\arpeggio

  \arpeggioParenthesis
  <c e g c>2\arpeggio

  \arpeggioParenthesisDashed
  <c e g c>2\arpeggio

  \arpeggioNormal
  <c e g c>2\arpeggio
}
\end{verbatim}

Die \texttt{\textbackslash dash}--Eigenschaft der Arpeggioklammern werden von der \texttt{\textbackslash dash-definition}--Eigenschaft kontrolliert, die beschrieben ist in \cite{Legatobögen}, Seite 123.

Ein Arpeggio kann auch explizit ausgeschrieben werden, indem Überbindungsbögen benutzt werden. Für mehr Information siehe \cite{Bindebögen}, Seite 49.

**Vordefinierte Befehle**

\texttt{\arpeggio}, \texttt{\arpeggioArrowUp}, \texttt{\arpeggioArrowDown}, \texttt{\arpeggioNormal}, \texttt{\arpeggioBracket}, \texttt{\arpeggioParenthesis}, \texttt{\arpeggioParenthesisDashed}. 
Ausgewählte Schnipsel

Arpeggio über mehrere Systeme in anderen Kontexten

Arpeggio über mehrere Systeme können in anderen Kontexten als dem PianoStaff erstellt werden, wenn der Span_arpeggio engraver in den Score-Kontext eingefügt wird.

\new PianoStaff \relative c' <<
\set PianoStaff.connectArpeggios = ##t
\new Staff {
  <c e g c>4\arpeggio
  <g c e g>4\arpeggio
  <e g c e>4\arpeggio
  <c e g c>4\arpeggio
}
\new Staff {
  \clef bass
  \repeat unfold 4 {
    <c, c g>4\arpeggio
  }
}
>>

Arpeggio zwischen Systemen in einem Klaviersonystem erstellen

In einem Klaviersonystem (PianoStaff) ist es möglich, ein Arpeggio zwischen beiden Systemen zu verbinden, indem die PianoStaff.connectArpeggios-Eigenschaft gesetzt wird.

\score {
  \new ChoirStaff {
    \set Score.connectArpeggios = ##t
    <<
      \new Voice \relative c' {
        <c e>2\arpeggio
        <d f>2\arpeggio
        <c e>1\arpeggio
      }
      \new Voice \relative c {
        \clef bass
        <c g'>2\arpeggio
        <b g'>2\arpeggio
        <c g'>1\arpeggio
      }
    >>
  }

  \context {
    \Score
Kapitel 1: Musikalische Notation

Arpeggios zwischen unterschiedlichen Stimmen erzeugen

Ein Arpeggio kann zwischen Noten aus unterschiedlichen Stimmen auf demselben System gezogen werden, wenn der \texttt{Span\_arpeggio\_engraver} in den \texttt{Staff}-Kontext verschoben wird:

\begin{verbatim}
\new Staff \with {
    \consists "Span\_arpeggio\_engraver"
}
\relative c' {
    \set Staff.connectArpeggios = ##t
    <<
        \{ <e' g>4\arpeggio <d f> <d f>2 \}
        \|
        \{ <d, f>2\arpeggio <g b>2 \}
    >>
}
\end{verbatim}

Siehe auch

- Glossar: Abschnitt “arpeggio” in Glossar.
- Notationsreferenz: [Legatobögen], Seite 123, [Bindebögen], Seite 49.

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 (\texttt{PianoStaff}) zu benutzen.

Die Arpeggios im Klammer-Stil funktionieren nicht über mehrere Notensysteme.

Triller

Kurze Triller ohne eine Dauer werden mit dem Befehl \texttt{\trill} notiert, siehe auch [Artikulationszeichen und Verzierungen], Seite 113.

Längere Triller mit einer Dauer werden mit den Befehlen \texttt{\startTrillSpan} zu Beginn und \texttt{\stopTrillSpan} am Ende erstellt.

\begin{verbatim}
\relative {
\end{verbatim}
Ein Triller-Strekcer, der über einen Zeilenumbruch geht, beginnt genau über der ersten Note auf der neue Zeile erneut.

Aufeinanderfolgende Trillerstrecker funktionieren ohne einen \stopTrillSpan-Befehl, weil ein folgender Strecker automatisch die rechte Begrenzung des vorhergehenden beendet.

Triller können auch mit Vorschlägen kombiniert werden. Die Syntax für diese Konstruktion und die Methode, um die Position der Vorschläge präzise zu positionieren, wird gezeigt in [Verzierungen], Seite 105.
Triller, die auf einer bestimmten Note ausgeführt werden sollen, können mit dem Befehl \pitchedTrill notiert werden. Das erste Argument ist die Hauptnote, das zweite die Note, auf der getrillert wird. Sie wird als Note ohne Hals in Klammern ausgegeben.

\relative {
  \pitchedTrill
  d''2\startTrillSpan fis
d2
c2\stopTrillSpan
  r2
}

Aufeinanderfolgende Versetzungszeichen der selben Note im selben Takt müssen selbst hinzugefügt werden. Nur das Versetzungszeichen des ersten Trillers mit Tonhöhe innerhalb eines Taktes wird ausgegeben.

\relative {
  \pitchedTrill
  eis''4\startTrillSpan fis
eis4\stopTrillSpan
  \pitchedTrill
eis4\startTrillSpan cis
eis4\stopTrillSpan
  \pitchedTrill
eis4\startTrillSpan fis
eis4\stopTrillSpan
  \pitchedTrill
eis4\startTrillSpan fis!
eis4\stopTrillSpan
}

Vordefinierte Befehle

\startTrillSpan, \stopTrillSpan.

Siehe auch

Glossar: Abschnitt “trill” in Glossar.
Notationsreferenz: [Artikulationszeichen und Verzierungen], Seite 113, [Verzierungen], Seite 105.
Referenz der Interna: Abschnitt “TrillSpanner” in Referenz der Interna.
1.4 Wiederholungszeichen

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

volta (Wiederholungsklammer)
Die wiederholte Musik wird nicht geschrieben, sondern zwischen zwei Wiederholungstaktstrichen eingeschlossen. Wenn die Wiederholung am Anfang eines Stückes beginnt, wird nur am Ende der Wiederholung eine Wiederholungstaktlinie gesetzt. Alternative Schlüsse (Volta) werden von links nach rechts mit Klamern gesetzt. Das ist die Standardnotationspraxis für Wiederholungen mit alternativen Schlüssen.

unfold (aufklappen)
Die wiederholte Musik wird ausgeschrieben, so oft, wie es durch Wiederholungszähler definiert wird. Das erspart Arbeit, wenn repetitive Musik notiert wird.

percent (Prozent-Wiederholung)
Das sind Noten- oder Taktwiederholungen, sie sehen aus wie ein Schrägstrich bzw. wie ein Prozentzeichen.

tremolo
Das wird benutzt, um Tremolo-Wiederholungen am Notenhals zu notieren.

1.4.1 Lange Wiederholungen

Normale Wiederholungen
Die Syntax für normale Wiederholungen ist

```
\repeat Typ Wiederholungszähler musikAusdr
```

wobei `musikAusdr` ein musikaler Ausdruck ist.

Wiederholung ohne alternativen Schluß:

```
\relative {  
  \repeat volta 2 \{ c'4 d e f \}  
c2 d
```

Alternative Schlüsse können mit `\alternative` gesetzt werden. Damit die alternativen Schlüsse von den wiederholten Noten abgegrenzt werden, müssen sie in geschweiften Klammern zusammengefasst werden.

\repeat volta Wiederholungszähler musikAusdr
\alternative {
  { musikAusdr }
}

wobei `musikAusdr` ein musikalischer Ausdruck ist.

Wenn es mehr Wiederholungen gibt, als Alternativen angegeben sind, erhalten die ersten Wiederholungen den ersten Schluss.

Eine einfache Wiederholung mit einer Alternative:

\relative {
  \repeat volta 2 { c''4 d e f | }
  \alternative {
    { c2 e | }
    { f2 g | }
  }
  c1
}

Eine einfache Wiederholung mit mehr als einer Alternative:

\relative {
  \repeat volta 4 { c''4 d e f | }
  \alternative {
    { c2 e | }
    { f2 g | }
  }
  c1
}

Mehrfache Wiederholungen mit mehr als einer Alternative:

\relative {
  \repeat volta 3 { c''4 d e f | }
  \alternative {
    { c2 e | }
  }
}
Kapitel 1: Musikalische Notation

Achtung: Wenn es zwei oder mehr Alternativen gibt, darf nichts zwischen der schließenden Klammer der einen und der öffnenden Klammer der anderen Wiederholung stehen, weil sonst nicht die erwartete Anzahl von Endungen produziert wird.


Wenn eine Wiederholung mitten in einem Takt beginnt und keine Alternativen hat, fällt normalerweise auch das Ende der Wiederholung mitten in einen Takt, sodass beide unvollständigen Takt einen vollständigen Takt ergeben. In diesem Fall bezeichnen die Wiederholungsstriche keine richtigen Taktstriche. Benutzen Sie nicht \partial-Befehle oder Taktüberprüfung, wo die Wiederholungslinien gesetzt werden:

\relative { % no \partial here
c'4 e g % no bar check here
% no \partial here
\repeat volta 4 {
e4 |
c2 e |
% no \partial here
g4 g g % no bar check here
}
% no \partial here
g4 |
a2 a |
g1 |
}

Ähnlich ist es, wenn eine Wiederholung mit einem Auftakt beginnt und keine Alternativen hat. In diesem Fall muss man aber den \partial-Befehl zu Beginn der Partitur setzen:

\relative {
\partial 4 % required
\repeat volta 4 {

Wenn alternative Endungen zu einer Wiederholung hinzugefügt werden, die mit einem Auf- 
takt beginnt, muss die \texttt{Timing.measureLength}-Eigenschaft manuell gesetzt werden, und an 
folgenden Stellen:

- am Beginn eines unvollständigen Taktes innerhalb der \texttt{alternative}-Umgebung, die nor-
  malerweise am Ende jeder Alternative auftreten, außer (in den meisten Fällen) in der letzten.
- zu Beginn jeder Alternative außer der ersten.

\begin{verbatim}
\partial 4
\repeat volta 2 { e4 | c2 e | }
\alternative {
  { 
    \set Timing.measureLength = #(ly:make-moment 3/4)
    g4 g g \% optional bar check is allowed here 
  }
  { 
    \set Timing.measureLength = #(ly:make-moment 4/4)
    a2 a | 
  }
}
g1 |
\end{verbatim}

Die \texttt{measureLength}-Eigenschaft ist beschrieben in [Verwaltung der Zeiteinheiten], Seite 111.

Der \texttt{\textbackslash inStaffSegno}-Befehl kann angewandt werden, um das Segno-Zeichen in die Notenzeile 
einzubinden, auch in Kooperation mit dem \texttt{\textbackslash repeat volta}-Befehl. Die benutzten Taktstreich-
symbole können durch Überschreiben der Eigenschaften \texttt{segnoType}, \texttt{startRepeatSegnoType}, 
\texttt{endRepeatSegnoType} bzw. \texttt{doubleRepeatSegnoType} geändert werden.

\begin{verbatim}
e1
\repeat volta 2 {
  \inStaffSegno
  f2 g a b
}
\end{verbatim}
Kapitel 1: Musikalische Notation

Ausgewählte Schnipsel

Volta-Klamern verkürzen

Volta-Klamern werden normalerweise über alle Noten der Klammer gezogen, aber es ist möglich sie zu verkürzen. Hierzu muss voltaSpannerDuration definiert werden, in dem Beispiel etwa als 3/4, sodass die Klammer nur einen Takt dauert.

Volta-Klamern zu zusätzlichen Systemen hinzufügen

Der Volta_engraver befindet sich im Score-Kontext und Klammern werden deshalb nur auf dem obersten System dargestellt. Das kann umgangen werden, indem man den Volta_engraver zu dem Staff-Kontext hinzufügt, in dem die Klammern zusätzlichen vorkommen sollen. Siehe auch das "Volta multi staff"-Schnipsel.
Setting the double repeat default for volte

There are three different styles of double repeats for volte, that can be set using `doubleRepeatType`.

```latex
\relative c' {
\repeat volta 1 { c1 }
\set Score.doubleRepeatType = #'":..:" 
\repeat volta 1 { c1 }
\set Score.doubleRepeatType = #'":|.|:" 
\repeat volta 1 { c1 }
\set Score.doubleRepeatType = #'":|.:"
\repeat volta 1 { c1 }
}
```

Alternative Taktnummerierung

Zwei alternative Methoden können eingestellt werden, die die Taktnummerierung beeinflussen, insbesondere bei Wiederholungen.

```latex
\relative c' {
\set Score.alternativeNumberingStyle = #'numbers 
\repeat volta 3 { c4 d e f | }
\alternative {
\set Score.alternativeNumberingStyle = #'numbers-with-letters
\repeat volta 3 { c4 d e f | }
\alternative {
{ c4 d e f | c2 d \break }
{ f4 g a b | f4 g a b | f2 a | \break }
{ c4 d e f | c2 d }
}
\repeat volta 3 { c4 d e f | }
\set Score.alternativeNumberingStyle = #'numbers-with-letters
\break}
```
Siehe auch

Glossar: Abschnitt “repeat” in Glossar, Abschnitt “volta” in Glossar.

Notationsreferenz: [Taktstriche], Seite 92, Abschnitt 5.1.4 [Umgebungs-Plugins verändern], Seite 558, [Verwaltung der Zeiteinheiten], Seite 111.

Schnipsel: Abschnitt “Repeats” in Schnipsel.

Bekannte Probleme und Warnungen

Bindebögen, die von einer \repeat-Umgebung in eine \alternative-Umgebung ragen, funktionieren nur in der ersten Klammer. Bindebögen können auch nicht von der Ende einer Wiederholungsklammer auf den Anfang der Wiederholung verweisen.

Wenn eine Wiederholung innerhalb eines unvollständigen Taktes beginnt und eine \alternative-Umgebung mit einer Veränderung von measureLength enthält, führt die Verwendung von \unfoldRepeats zu falsch gesetzten Taktstrichen und Taktüberprüfungswarnungen.

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.

Manuelle Wiederholungszeichen

Achtung: Diese Methoden werden nur verwendet, um ungewöhnliche Wiederholungskonstruktionen darzustellen und können sich unerwünscht verhalten. In den meisten Fällen sollten Wiederholungen mit dem Befehl \repeat erstellt werden oder indem die entsprechenden Taktstriche eingegeben werden. Mehr Information in [Taktstriche], Seite 92.

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 \relative{Taktlinie}.
\relative{
c"1
\set Score.repeatCommands = #'(start-repeat)
d4 e f g
c1}

\noteheads.s0/\noteheads.s2/\timesig.C44/c1/noteheads.s0/\noteheads.s2

Der Notensatzpraxis folgend werden Wiederholungstaktstriche nicht zu Beginn eines Stückes gesetzt.

end-repeat
Setzt eine \relative{Taktlinie}.
\relative{
c"1
d4 e f g
\set Score.repeatCommands = #'(end-repeat)
(volta Zahl) ... (volta #f)
Setzt eine Volta-Klammer mit der Beschriftung Nummer. Die Volta-Klammer muss explizit beendet werden, sonst wird sie nicht ausgegeben.

```
\relative {
    \set Score.repeatCommands = #'((volta "2"))
    g4 a g a
    \set Score.repeatCommands = #'((volta #f))
    c1
}
```

Mehrfache Wiederholungszeichen können an der selben Stelle vorkommen:

```
\relative {
    \set Score.repeatCommands = #'((volta "2, 5") end-repeat)
    g4 a g a
    \set Score.repeatCommands = #'((volta #f) (volta "95") end-repeat)
    b1
    \set Score.repeatCommands = #'((volta #f))
}
```

Text kann auch in der Volta-Klammer gesetzt werden. Der Text kann aus Zahlen oder einer Zahl oder einer Textbeschriftung bestehen, siehe Abschnitt 1.8.2 [Text formatieren], Seite 223. Die einfachste Art Text zu benutzen ist, die Beschriftung zuerst zu definieren und dann die Beschriftung in einer Scheme-Liste einzufügen.

```
voltaAdLib = \markup { 1. 2. 3... \text \italic { ad lib. } }
\relative {
    c''1
    \set Score.repeatCommands =
        #(list(list 'volta voltaAdLib) 'start-repeat)
    c4 b d e
    \set Score.repeatCommands = #'((volta #f) (volta "4.")) end-repeat)
    f1
    \set Score.repeatCommands = #'((volta #f))
```
Ausgeschriebene Wiederholungen

Mit dem \texttt{unfold}-Befehl können Wiederholungen eingesetzt werden, um repetitive Musik zu notieren. Die Syntax ist
\begin{verbatim}
\repeat unfold Wiederholungszähler musikAusdr
\end{verbatim}

wobei \texttt{musikAusdr} ein musikalischer Ausdruck ist und \texttt{Wiederholungszähler} die Anzahl bezeichnet, mit der \texttt{musikAusdr} wiederholt wird.

\begin{verbatim}
\relative {
  \repeat unfold 2 { c''4 d e f }
  c1
}
\end{verbatim}

In einigen Fällen, insbesondere in einer \texttt{relative}-Umgebung, bedeutet die Funktion \texttt{\repeat unfold} nicht das gleiche wie die ausgeschriebenen Noten mehrere Male. Beispielsweise ist
\begin{verbatim}
\repeat unfold 2 { a'4 b c }
\end{verbatim}

nicht das Selbe wie
\begin{verbatim}
a'4 b c | a'4 b c
\end{verbatim}

Repetitive Wiederholungen können auch mit mehreren Alternativeklammern notiert werden:
\begin{verbatim}
\relative {
  \repeat unfold 2 { c''4 d e f }
  \alternative {
    { c2 g' }
    { c,2 b }
  }
  c1
}
\end{verbatim}
Wenn es mehr Wiederholungen als Alternativen gibt, wird die erste Alternative so oft eingesetzt, bis sich zusammen mit den restlichen Alternativen die Gesamtanzahl der Wiederholungen ergibt.

Wenn es mehr Alternativen als Wiederholungen gibt, wird nur die ersten Alternativen ausgegeben und die restlichen Alternativen ignoriert und nicht gesetzt.

Es ist auch möglich, mehrere unfold-Wiederholungen (mit oder ohne Alternativen) ineinander zu verschachteln:

Akkordkonstruktionen können mit dem Akkordwiederholungssymbol q wiederholt werden. Siehe [Akkord-Wiederholungen], Seite 154.

Siehe auch

Notationsreferenz: [Akkord-Wiederholungen], Seite 154.
Schnipsel: Abschnitt “Repeats” in Schnipsel.

1.4.2 Kurze Wiederholungen


Prozent-Wiederholungen

Kurze wiederholte Muster werden einmal gesetzt und das wiederholte Muster wird durch ein besonderes Zeichen ersetzt.

Die Syntax lautet:
\repeat percent Wiederholungszahl musikAusdr
wobei musikAusdr ein musikalischer Ausdruck ist.

Muster, die kürzer als ein Takt sind, werden mit Schrägstrichen ersetzt:
\relative c' {
  \repeat percent 4 { c128 d e f }
  \repeat percent 4 { c64 d e f }
  \repeat percent 5 { c32 d e f }
  \repeat percent 4 { c16 d e f }
  \repeat percent 4 { c8 d }
  \repeat percent 4 { c4 }
  \repeat percent 2 { c2 }
}

Muster von einem oder zwei Takten Dauer werden mit prozentartigen Symbolen ersetzt:
\relative c' {
  \repeat percent 2 { c4 d e f }
  \repeat percent 2 { c2 d }
  \repeat percent 2 { c1 }
}
Muster, die kürzer als ein Takt sind, aber unterschiedliche Dauern beinhalten, benützen ein doppeltes Prozent-Symbol.

Ausgewählte Schnipsel

*Prozent-Wiederholungen zählen*

Ganzaktwiederholungen mit mehr als zwei Wiederholungen erhalten einen Zähler, wenn man die entsprechende Eigenschaft einsetzt:

```
\relative c'' {
    \set countPercentRepeats = ##t
    \repeat percent 4 { c1 }
}
```

*Sichtbarkeit von Prozent-Wiederholungen*

Prozentwiederholungszähler können in regelmäßigen Intervallen angezeigt werden, indem man die Eigenschaft `repeatCountVisibility` beeinflusst.

```
\relative c'' {
    \set countPercentRepeats = ##t
    \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
    \repeat percent 10 { c1 } \break
    \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
    \repeat percent 6 { c1 d1 }
}
```
Isolierte Prozentwiederholungen
Isolierte Prozentwiederholungen können auch ausgegeben werden. Das wird erreicht, indem man eine Ganzaktupause notiert und ihre Ausgabeform ändert:

```lisp
(makePercent =
  #(define-music-function (note) (ly:music?)
    "Make a percent repeat the same length as NOTE."
    (make-music 'PercentEvent
      'length (ly:music-length note)))

\relative c'' {
  \makePercent s1
}
```

Siehe auch
Glossar: Abschnitt “percent repeat” in Glossar, Abschnitt “simile” in Glossar.
Schnipsel: Abschnitt “Repeats” in Schnipsel.

Tremolo-Wiederholung
Tremolos können in zwei Arten notiert werden: als Wechsel zwischen zwei Noten oder Akkorden oder als schnelle Wiederholung einer einzigen Note. Tremolos, die als Wechsel realisiert werden, werden dargestellt, indem Balken zwischen die Noten gesetzt werden, Tremolos, die eine schnelle Wiederholung darstellen, haben Balken oder Schrägstriche am Hals einer einzigen Note.
Um Tremolobalken zwischen Noten zu setzen, kann der \repeat-Befehl mit dem Tremolo-Stil benutzt werden:

```latex
\relative c'' {
  \repeat tremolo 8 { c16 d }
  \repeat tremolo 6 { c16 d }
  \repeat tremolo 2 { c16 d }
}
```

Die \repeat tremolo-Syntax braucht genau zwei Noten innerhalb der geschweiften Klammern, und die Anzahl der Wiederholungen muss einem Wert entsprechen, der mit einfachen oder punktierten Noten ausgedrückt werden kann. \repeat tremolo 7 funktioniert und setzt Tremolo für die Dauer einer Doppelpunktierten, aber \repeat tremolo 9 funktioniert nicht.
Die Dauer des Tremolos entspricht der Dauer der Wertes in Klammern, multipliziert mit der Zahl der Wiederholungen: \(\texttt{\textbackslash repeat tremolo 8 \{ c16 d16 \}}\) ergibt ein Tremolo für eine Ganze, notiert als zwei Ganze, die zwei Tremolobalken zwischen sich haben.

Es gibt zwei Möglichkeiten, ein Tremolozeichen zu einer einzelnen Noten hinzuzufügen. Die \(\texttt{\textbackslash repeat tremolo}\)-Syntax kann hier auch benutzt werden; in diesem Fall wird die Note allerdings nicht eingeklammert:

\[\texttt{\textbackslash repeat tremolo 4 c'16} \]

\[\text{\begin{music}
\relative {
c''2:8 c:32
\line c: c:
\}
}\end{music}}\]

Ausgewählte Schnipsel

Cross-staff tremolos

Since \(\texttt{\textbackslash repeat tremolo}\) expects exactly two musical arguments for chord tremolos, the note or chord which changes staff within a cross-staff tremolo should be placed inside curly braces together with its \(\texttt{\textbackslash change Staff}\) command.

\[\texttt{\textbackslash new PianoStaff << \new Staff = "up" \relative c' \{\key a \major \time 3/8 s4.\}\new Staff = "down" \relative c' \{\key a \major \time 3/8 \voiceOne \repeat tremolo 6 \{<a e'>32 \{ \change Staff = "up" \voiceTwo <cis a' dis>32 \}\}\}\}}\]
1.5 Gleichzeitig erscheinende Noten


1.5.1 Eine einzelne Stimme

Dieser Abschnitt behandelt gleichzeitige Noten innerhalb derselben Stimme.
Kapitel 1: Musikalische Notation

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, genau so wie bei einfachen Noten.

\relative {
  <a' c e>1 <a c e>2 <f a c e>4 <a c>8. <g c e>16
}

Akkorde können auch von Artikulationen gefolgt werden, genau wie auch einfache Noten.

\relative {
  <a' c e>1\fermata <a c e>2-> <f a c e>4\prall <a c>8.^! <g c e>16-.
}

Die Noten innerhalb der Akkorde können auch von Artikulationen oder Ornamenten gefolgt werden.

\relative {
  <a'\prall c e>1 <a--> c\turn e>2 <f-. a c-. e-.>4
  <a-+ c-->8. <g\fermata c e\turn>16
}

Manche Notationselemente, wie etwa Dynamik, Crescendo-Klammern und Legatobögen müssen an den gesamten Akkord gehängt werden und nicht an einzelne Noten, damit sie ausgegeben werden.

\relative {
  <a'\f c( e>1 <a c) e>\f <a\< c e)( <a\! c e>)
  <a c e>\< <a c e> <a c e>\!
}

Ein Akkord ist sozusagen ein Container für die Noten, ihre Artikulationen und andere angehängte Elemente. Demzufolge hat also auch ein Akkord ohne wirkliche Noten innerhalb der Klammern keine Dauer. Alle angehängten Artikulationen geschehen zur selben musikalischen Zeit wie die folgende Note oder der folgende Akkord und werden damit kombiniert (für
komplexere Möglichkeiten, derartige Elemente zu kombinieren, siehe [Gleichzeitige Ausdrücke], Seite 155.

\relative {
  \grace { g'8( a b )< > ) \p \< -. \markup \italic "sempre staccato"
  \repeat unfold 4 { c4 e } c1\f}

Der relative Modus kann auch für Tonhöhen in Akkorden eingesetzt werden. Die erste Note eines Akkordes ist immer relativ zur ersten Note des vorherigen Akkordes, oder mit der Tonhöhe der letzten Note vor dem Akkord (wenn kein Akkord vorhergeht). Alle anderen Noten innerhalb des Akkordes sind relativ zu der Note vorher innerhalb des selben Akkordes.

\relative {
  <a' c e>1 <f a c> <a c e> <f' a c> <b, e b,>}

Mehr Information über Akkorden findet sich in Abschnitt 2.7 [Notation von Akkorden], Seite 387.

Siehe auch
Musikglossar: Abschnitt “chord” in Glossar.
Handbuch zum Lernen: Abschnitt “Noten zu Akkorden verbinden” in Handbuch zum Lernen.
Notationsreferenz: Abschnitt 2.7 [Notation von Akkorden], Seite 387, [Artikulationszeichen und Verzierungen], Seite 113, [Relative Oktavenbezeichnung], Seite 2, Abschnitt 1.5.2 [Mehrere Stimmen], Seite 157.
Schnipsel: Abschnitt “Simultaneous notes” in Schnipsel.

Bekannte Probleme und Warnungen
Akkorde, die mehr als zwei Tonhöhen für einen Notenlinenzwischenraum enthalten (wie etwa '<e f! fis!>') produzieren überlappende Notenköpfe. Abhängig von der Situation kann eines der folgenden Dinge helfen, die Darstellung zu verbessern:

• Kurzzeitig mehrere Stimmen benutzen, siehe Abschnitt 1.5.2 [Mehrere Stimmen], Seite 157: '<y f ! \< e fis!> >>'.
• enharmonische Transkription für einen oder mehrere Tonhöhen vornehmen: '<e f ges>' oder
• Cluster, siehe [Cluster], Seite 157.
Akkord-Wiederholungen

Um Schreibarbeit zu ersparen, kann ein Zeichen benutzt werden, um den vorhergehenden Akkord zu wiederholen. Das Symbol hierzu ist q:

\begin{verbatim}
\relative {
  <a' c e>1 q <f a c>2 q
}
\end{verbatim}


\begin{verbatim}
\relative {
  <a' c e>1\p^"text" q2\<\( q8)\[-! q8.]\! q16-1-2-3 q8\prall
}
\end{verbatim}

Das Akkordwiederholungssymbol erinnert sich an das letzte Vorkommen eines Akkordes, sodass man den letzten Akkord wiederholen kann, auch wenn in der Zwischenzeit nicht-Akkord-Noten oder -Pause auftreten sind.

\begin{verbatim}
\relative {
  <a' c e>1 c'4 q2 r8 q8 |
  q2 c, |
}
\end{verbatim}

Das Akkord-Wiederholungssymbol behält keine Dynamikzeichen, Artikulationen oder Ornamente, die in oder an den vorhergehenden Akkord gehängt waren.

\begin{verbatim}
\relative {
  <a-. c\prall e>1\sfz c'4 q2 r8 q8 |
  q2 c, |
}
\end{verbatim}

Damit auch diese Zeichen erhalten bleiben, kann die Funktion \chordRepeats explizit mit einem zusätzlichen Argument aufgerufen werden, um eine Liste an Ereignistypen (engl. event) zu spezifizieren, die mit wiederholt werden, es sei denn, der gleiche Ereignistyp wird selber mit dem q verwendet:

\begin{verbatim}
\relative {

Die Benutzung von `\chordRepeats` innerhalb einer `\relative`-Konstruktion führt zu unerwarteten Ergebnissen: Nachdem die Akkordereignisse einmal erweitert sind, können sie nicht mehr von normal eingegebenen Akkorden unterschieden werden. Dadurch fügt `\relative` einen Oktavsprung entsprechend des aktuellen Kontexts ein.

Weil geschachtelte `\relative`-Umgebungen sich nicht gegenseitig beeinflussen, kann man eine zusätzliche `\relative`-Umgebung innerhalb von `\chordRepeats` benutzen, um die Oktavbeziehungen darzustellen, bevor die wiederholten Akkorde eingesetzt werden. In diesem Fall beeinflusst der gesamte Inhalt der inneren `\relative`-Umgebung nicht die äußere. Daraus ergibt sich die unterschiedliche Oktave der letzten Note in diesem Beispiel:

```
\new Voice
\relative c'' {
  \chordRepeats #'(articulation-event)
  \relative
  { <a'-. c\prall e>1\sfz c'4 q2 r8 q8-. } |
  q2 c, |
}
```

Derartige Probleme mit `\relative` treten nur auf, wenn `\chordRepeats` explizit aufgerufen wird: die Verarbeitung von einfachem `q` wird erst vorgenommen, wenn alle `\relative`-Umgebungen schon verarbeitet sind.

Siehe auch

Notationsreferenz: Abschnitt 2.7 [Notation von Akkorden], Seite 387, [Artikulationszeichen und Verzierungen], Seite 113.


Gleichzeitige Ausdrücke

Eine oder mehrere musikalische Ausdrücke, die in doppelte spitze Klammern eingeschlossen werden, werden gleichzeitig gesetzt. Wenn der erste Ausdruck mit einer einzelnen Note beginnt oder die gesamte Konstruktion explizit in einer einzelnen Stimme erstellt wird, wird auch nur ein Notensystem erstellt. In anderem Falle werden die Elemente der simultanen Konstruktion auf unterschiedlichen Systemen gesetzt.
Das nächste Beispiel zeigt simultane Konstruktionen auf einem System:

\new Voice { \% explicit single voice
   << \relative { a'4 b g2 }
      \relative { d'4 g c,2 } >>
}

\relative {
   % single first note
   a' << \relative { a'4 b g }
      \relative { d'4 g c, } >>
}

Dass kann benutzt werden, wenn die simultanen Abschnitte einen identischen Rhythmus haben, aber wenn versucht wird, Noten mit unterschiedlicher Dauer an denselben Hals zu setzen, gibt es Fehlermeldungen. Noten, Artikulationen und Eigenschaftsänderungen in einer einzelnen 'Voice' werden gesammelt und in musikalischer Reihenfolge gesetzt:

\relative {
   <a' c>4-. <>-. << c a >> << { c-. <c a> } { a s-. } >>
}

Mehrfache Hälse oder Balken oder unterschiedliche Notendauern oder Eigenschaften zur selben musikalischen Zeit erfordern den Einsatz von mehreren Stimmen.

Das nächste Beispiel zeigt, wie ein simultaner Ausdruck implizit mehrere Systeme erstellt:

\relative {
   % no single first note
   << \relative { a'4 b g2 }
      \relative { d'4 g2 c,4 } >>
}

In diesem Fall stellt der unterschiedliche Rhythmus kein Problem dar, weil sie in unterschiedlichen Stimmen interpretiert werden.
Bekannte Probleme und Warnungen

Wenn Noten zweier oder mehrerer Stimmen mit Hälse in die gleiche Richtung an der selben Position auf dem System gesetzt werden und keinen Versatz durch `shift` aufweisen (oder den gleichen Versatz besitzen), erscheint die Nachricht

WARNUNG: zu viele kollidierende Notenspalten werden ignoriert

während der Kompilation. Diese Nachricht kann unterdrückt werden durch

```
\override NoteColumn.ignore-collision = ##t
```

Das unterdrückt jedoch nicht nur die Warnungen, sondern schaltet auch die Auflösung von Zusammenstößen ab und kann also zu unbeabsichtigten Resultaten führen. (Siehe auch [Auflösung von Zusammenstößen], Seite 161.)

Cluster

Ein Cluster zeigt an, dass alle Tonhöhen in einem Bereich gleichzeitig gespielt werden sollen. Cluster können gedeutet werden als eine Zusammenfassung einer ganzen Anzahl von Noten. Sie werden notiert, indem die Funktion `\makeClusters` auf eine Reihe von Akkorden angewendet wird:

```
\relative \makeClusters { <g' b>2 <c g'> }
```

Normale Noten und Cluster können zusammen im selben System notiert werden, sogar gleichzeitig. In solchen Fällen wird nicht versucht, automatisch Zusammenstöße zwischen normalen Noten und Clustern aufzulösen.

Siehe auch

Musikglossar: Abschnitt “cluster” in Glossar.
Schnipsel: Abschnitt “Simultaneous notes” in Schnipsel.

Bekannte Probleme und Warnungen

Cluster sehen nur gut aus, wenn sie wenigstens über zwei Akkorde reichen – andernfalls sind sie zu schmal.

Cluster haben keine Hälse und können auch selber keine Dauern darstellen, aber die Länge des gesetzten Clusters wird erschlossen anhand der Dauern der definierten Akkorde. Voneinander getrennte Cluster brauchen eine unsichtbare Pause zwischen sich.

Cluster produzieren kein MIDI.

1.5.2 Mehrere Stimmen

Dieser Abschnitt behandelt gleichzeitige Noten in mehreren Stimmen oder mehreren Systemen.

Mehrstimmigkeit in einem System

*Stimmen explicit beginnen*

Die grundlegende Struktur, die man benötigt, um mehrere unabhängige Stimmen in einem Notensystem zu setzen, ist im Beispiel unten dargestellt:

```
\new Staff <<
```
Stimmen werden hier explizit erstellt und erhalten Bezeichnungen zugewiesen. Die \voiceOne ...
\voiceFour-Befehle stellen die Stimmen so ein, dass für die erste und dritte Stimme die Häse nach oben zeigen, für die zweite und vierte Stimme hingegen nach unten. Die Noten der dritten und vierten Stimme werden horizontal verschoben, und Pausen in den entsprechenden Stimmen werden automatisch verschoben, um Zusammenstöße zu vermeiden. Der \oneVoice-Befehl stellt das Standardverhalten mit neutralen Halsrichtungen wieder her.

Vorübergehende polyphone Passagen

Ein vorübergehender polyphoner Abschnitt kann mit folgender Konstruktion erstellt werden:

\begin{verbatim}
<< { \voiceOne ... }
\new Voice \{ \voiceTwo ... \}
>> \oneVoice
\end{verbatim}

Der erste Ausdruck innerhalb des polyphonen Abschnitts wird in den Voice-Kontext gestellt, der unmittelbar vor dem polyphonen Abschnitt aktiv war, und der gleiche Voice-Kontext setzt sich nach dem Abschnitt fort. Andere Ausdrücke innerhalb der eckigen Klammern werden an-
deren Stimmennummern zugewiesen. Damit lassen sich auch Gesangstexte einer durchgehenden Stimme vor, während und nach dem polyphonen Abschnitt zuweisen:

\begin{verbatim}
\relative <<
\new Voice = "melody" \{ 
  a'4
<<
  \{ 
    \voiceOne 
    g f 
  \}
\new Voice \{ 
  \voiceTwo 
  d2 
\}
>> 
\oneVoice 
  e4
\} 
\new Lyrics \lyricsto "melody" \{ 
  This is my song.
\}
\end{verbatim}
Hierbei sind die Befehle \voiceOne und \voiceTwo notwendig, um die Einstellungen für jede Stimme zu initialisieren.

Die Konstruktion mit doppeltem Backslash


Das erste Beispiel könnte also wie folgt notiert werden:

```latex
<< \relative { r8 r16 g'' e8. f16 g8[ c,] f e16 d } \n\relative { d''16 c d8~ 16 b c8~ 16 b c8~ 16 b8. } >>
```

Diese Syntax kann benutzt werden, wenn es keine Rolle spielt, ob vorübergehend Stimmen erstellt werden und dann wieder verworfen werden. Diese implizit erstellten Stimmen erhalten die Einstellungen, die in den Befehlen \voiceOne ... \voiceFour enthalten sind, in der Reihenfolge, in der sie im Quelltext auftauchen.

Im nächsten Beispiel zeigen die Hälfte der zeitweiligen Stimme nach oben, sie wird deshalb erst als dritte in der Konstruktion notiert, damit sie die Eigenschaften von voiceThree zugewiesen bekommt. Unsichtbare Pause werden eingesetzt, damit keine doppelten Pausen ausgegeben werden.

```latex
<< \relative { r8 g'' g g f16 ees f8 d } \n\relative { ees'8 r ees r d r d r } \n\relative { d''8 s c s bes s a s } >>
```

Es wird sehr empfohlen, in allen außer den allereinfachsten Stücken explizite Stimmenkontexte zu erstellen, wie erklärt in Abschnitt “Kontexte und Engraver” in Handbuch zum Lernen und Abschnitt “Stimmen explizit beginnen” in Handbuch zum Lernen.

Stimmen-Anordnung

Wenn mehrere Stimmen notiert werden, sollte folgende Anordnung eingehalten werden:

Stimme 1: höchste
Kapitel 1: Musikalische Notation

Stimme 2: tiefste
Stimme 3: zweithöchste
Stimme 4: zweittiefste
Stimme 5: dritthöchste
Stimme 6: dritttiefste

usw.

Auch wenn das erst nicht einleuchtend erscheint, erleichtert es den automatischen Layoutprozess doch sehr. Die ungeraden Stimmen erhalten Hälse nach oben, die graden Stimmen Hälse nach unten:

\new Staff <<
\time 2/4
\{ f''2 \} % 1: highest
\\%
\{ c'2 \} % 2: lowest
\\%
\{ d''2 \} % 3: second-highest
\\%
\{ e'2 \} % 4: second-lowest
\\%
\{ b'2 \} % 5: third-highest
\\%
\{ g'2 \} % 6: third-lowest
>>

\begin{center}
\includegraphics[width=2cm]{staff.png}
\end{center}

\begin{center}
\begin{box}
\textbf{Achtung:} Gesangstext und Strecker (etwa Bögen, Bindebögen, Crescendoklammern usw.) können nicht zwischen zwei Stimmen erstellt werden.
\end{box}
\end{center}

Identische Rhythmen

Wenn parallele Abschnitte gesetzt werden sollen, die identischen Rhythmus haben, kann man die Ausdrücke in einen einzigen Voice-Kontext parallel kombinieren, sodass sich Akkorde ergeben. Um das zu erreichen, müssen sie einfach von spitzen Klammern innerhalb einer expliziten Stimme umgeben werden:

\new Voice <<
\relative { e''4 f8 d e16 f g8 d4 }
\relative { c''4 d8 b c16 d e8 b4 }
>>

\begin{center}
\includegraphics[width=2cm]{voice.png}
\end{center}

Mit dieser Methode können sich seltsame Balken und Warnungen ergeben, wenn die Musikausdrücke nicht den gleichen Rhythmus haben.

Vordefinierte Befehle

\voiceOne, \voiceTwo, \voiceThree, \voiceFour, \oneVoice.
Siehe auch
Notationsreferenz: [Schlagzeugsysteme], Seite 367, [Unsichtbare Pausen], Seite 54, [Hälse], Seite 210.
Schnipsel: Abschnitt “Simultaneous notes” in Schnipsel.

Stimmenstile
Stimmen können unterschiedliche Farben erhalten, um einfach erkennbar zu sein:

```
<< 
\relative { \voiceOneStyle d''4 c2 b4 } \\ 
\relative { \voiceTwoStyle e'2 e } \\ 
\relative { \voiceThreeStyle b2. c4 } \\ 
\relative { \voiceFourStyle g'2 g } 
>>
```

Der \voiceNeutralStyle-Befehl wird benutzt, um wieder die Standardausgabe einzuschalten.

Vordefinierte Befehle
\voiceOneStyle, \voiceTwoStyle, \voiceThreeStyle, \voiceFourStyle, \voiceNeutralStyle.

Siehe auch
Schnipsel: Abschnitt “Simultaneous notes” in Schnipsel.

Auflösung von Zusammenstößen

```
<< 
\relative { 
\relative { \relative { \relative { c''8 d e d c d c4 
\g'2 fis 
} \ \ } \ \ 
\relative { \relative { \relative { c''2 c8. b16 c4 
} \ \ } \ \ 
} \ \ 
} \ \ 
} 
```
Noten mit unterschiedlichen Notenköpfen können verschmolzen werden, mit der Ausnahme von Halben- und Viertelnotenköpfen, wie im Beispiel unten gezeigt. Hier werden die Notenköpfe auf Taktposition 1 im ersten Takt verschmolzen:

```latex
\relative { \oneVoice s1 e'8 a b c d2 }
```

Auch Köpfe mit unterschiedlichen Punktierungen wie auf Taktposition 3 im ersten Takt können verschmolzen werden:

```latex
\relative { \mergeDifferentlyHeadedOn \mergeDifferentlyDottedOn c''8 d e d c d c4 g'2 fis }
\relative { c''2 c8. b16 c4 e,2 r }
\relative { \oneVoice s1 e'8 a b c d2 }
```
Kapitel 1: Musikalische Notation

Die Halbe und die Achtel am Anfang des zweiten Taktes werden fehlerhaft verschmolzen, weil die automatische Verschmelzung nicht richtig arbeiten kann, wenn drei oder mehr Noten zur gleichen Zeit auftreten – und in diesem Fall ist der verschmolzene Notenkopf nicht richtig. Um das Verschmelzen zuzulassen, muss ein \shift (Verschiebung) auf die Note angewendet werden, die nicht verschmolzen werden soll. In diesem Fall wurde \shiftOn gesetzt, um das oberste g aus der Kolumne zu entfernen. Jetzt funktioniert \mergeDifferentlyHeadedOn (verschmelze Noten mit unterschiedlichen Köpfen) so wie es soll.

Der \shiftOn-Befehl ermöglicht die Noten einer Stimme zu verschieben, erzwingt dieses Verhalten aber nicht. Wenn \shiftOn auf eine Stimme angewendet wird, eine Note oder ein Akkord in der Stimme wird nur verschoben, wenn sein Hals mit dem Hals der Note einer anderen Stimme kollidieren würde, und nur, wenn der Hals der Kollisionsnote in die gleiche Richtung zeigt. Der \shiftOff-Befehl verhindert, dass eine derartige Verschiebung stattfinden kann.

Die äußeren Stimmen (also normalerweise Stimmen 1 und 2) haben als Standard \shiftOff eingestellt, während die inneren Stimmen (3 und mehr) \shiftOn definiert haben. Wenn eine Verschiebung stattfindet, werden Stimmen mit den Hälsern nach oben (also ungerade Stimmen) nach rechts verschoben, während Stimmen mit den Hülsen nach unten (also gerade Stimmen) nach links verschoben werden.
Hier ein Beispiel, das verstehen hilft, wie ein verkürzter polyphonischer Abschnitt intern ausgeweitet wird.

\begin{quote}
\textbf{Achtung:} Wenn Sie drei oder mehr Stimmen haben, sollte die vertikale Anordnung der Stimmen in der Eingabedatei nicht die gleiche sein wie die vertikale Anordnung der Stimmen im Notensystem!
\end{quote}

\begin{verbatim}
\new Staff \relative {
  \%% abbreviated entry
  \<<
    { f''2 } % 1: highest
    { g,2 } % 2: lowest
    { d'2 } % 3: upper middle
    { b2 } % 4: lower middle
  \>
  \%% internal expansion of the above
  \<<
    \new Voice = "1" { \voiceOne \shiftOff f'2 }
    \new Voice = "2" { \voiceTwo \shiftOff g,2 }
    \new Voice = "3" { \voiceThree \shiftOn d'2 } % shifts right
    \new Voice = "4" { \voiceFour \shiftOn b2 } % shifts left
  \>
}
\end{verbatim}

Zwei zusätzliche Befehle, \texttt{shiftOnn} und \texttt{shiftOnnn} stellen weitere Verschiebungsebenen zu Verfügung, die vorübergehend eingesetzt werden können um Zusammenstöße in kompliziertenSituationen aufzulösen. Siehe auch Abschnitt "Beispiel aus dem Leben" in Handbuch zum Lernen.

Noten werden nur verschmolzen, wenn ihre Hälse in entgegengesetzte Richtungen zeigen (also etwa wie Voice 1 und 2 in den Standardeinstellungen oder wenn die Hälse explizit in unterschiedliche Richtungen gedreht sind).

\textbf{Vordefinierte Befehle}

\texttt{mergeDifferentlyDottedOn, mergeDifferentlyDottedOff, mergeDifferentlyHeadedOn, mergeDifferentlyHeadedOff, shiftOn, shiftOff, shiftOnn, shiftOnnn, shiftOff.}

\textbf{Ausgewählte Schnipsel}

\textit{Zusätzliche Stimmen, um Zusammenstöße zu vermeiden}

Ein einigen Fällen von sehr komplexer polyphoner Musik sind zusätzliche Stimmen notwendig, um Zusammenstöße zwischen den Noten zu vermeiden. Wenn mehr als vierparallele Stimmen benötigt werden, können zusätzliche Stimmen definiert werden, indem eine Variable mit der Funktion \texttt{context-spec-music} definiert wird.

\begin{verbatim}
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)
\end{verbatim}
Horizontale Verschiebung von Noten erzwingen


\relative c' { <<
  \once \override NoteColumn.force-hshift = #1.7
  <d g>2 <d g>
}
Kapitel 1: Musikalische Notation

Siehe auch

Musikglossar: Abschnitt “polyphony” in Glossar.


Schnipsel: Abschnitt “Simultaneous notes” in Schnipsel.

Referenz der Interna: Abschnitt “NoteColumn” in Referenz der Interna, Abschnitt “NoteColumn ignore-collision” in Referenz der Interna, Abschnitt “RestCollision” in Referenz der Interna.

Bekannte Probleme und Warnungen

Die Benutzung von \override NoteColumn.ignore-collision = ##t führt dazu, dass Noten mit unterschiedlichen Köpfen in unterschiedlichen Stimmen falsch verschmolzen werden.

Automatische Kombination von Stimmen

Automatische Kombination von Stimmen wird verwendet, um zwei selbständige Stimmen auf einem Notensystem zu setzen. Es wird vor allem in Orchesterpartituren eingesetzt. Wenn beide Stimmen die gleichen Noten haben, wird nur eine Stimme gesetzt, wenn sie sich unterscheiden, werden sie als unterschiedliche Stimmen (Voice) gesetzt, und die Richtung der Häse wird automatisch bestimmt. Zusätzlich werden solo und a due-Stellen erkannt und bezeichnet.

Die Syntax zur automatischen Stimmenkombination lautet:

\partcombine musikAusdr1 musikAusdr2

Das nächste Beispiel zeigt, wie die Kombination funktioniert. Hier werden die Stimmen erst auf einem jeweils eigenen System und dann kombiniert gesetzt. Die gleichen Variablen werden für die Stimmen und das kombinierte System benutzt.

```
instrumentOne = \relative {
c'4 d e f |
R1 |
d'4 c b a |
b4 g2 f4 |
e1 |
}
```

```
instrumentTwo = \relative {
```

Beide Argumente von `\partcombine` werden als eigenständige `Voice`-Kontexte interpretiert. Wenn relative Oktaven benutzt werden, muss `\relative` für beide Stimmen benutzt werden, also:

```latex
\partcombine
  \relative ... musikAusdr1
  \relative ... musikAusdr2
```

Ein `\relative`-Abschnitt, der `\partcombine` umschließt, hat keinen Einfluss auf die Tonhöhen von `musikAusdr1` oder `musikAusdr2`.

In professionellen Partituren werden Stimmen oft für lange Strecken auseinander gehalten, auch wenn eine oder mehrere Noten tatsächlich aufeinander treffen und einfach als Unisono gesetzt werden könnten. Noten zu Akkorden zusammenzufassen oder eine Stimme als Solo anzuzeigen ist darum nicht ideal, weil die `\partcombine`-Funktion jede Note separat betrachtet. Aus diesem Grund kann die `\partcombine`-Funktion mit folgenden Befehlen verändert werden:

Befehle, die auf ... `Once` enden, gelten nur für die nächste Note eines musikalischen Ausdrucks.

- `\partcombineApart` und `\once \partcombineApart` erhalten die Noten als zwei unterschiedliche Stimmen, auch wenn sie als Akkord oder Unisono kombiniert werden könnten.
- `\partcombineChords` und `\once \partcombineChords` kombinieren die Noten als Akkord.
- `\partcombineUnisono` und `\once \partcombineUnisono` kombinieren beide Stimmen als Unisono.
Kapitel 1: Musikalische Notation

• \partcombineSoloI und \once \partcombineSoloI setzen nur Stimme eins und markieren sie als „Solo“.
• \partcombineSoloII und \once \partcombineSoloII setzen nur Stimme zwei und markieren sie als „Solo“.
• \partcombineAutomatic und \once \partcombineAutomatic beenden die Wirkung der Befehle oben und stellt das normale Verhalten des Kombinationsmechanismus wieder her.

\begin{verbatim}
instrumentOne = \relative c' {
  \partcombineApart c2^"apart" e |
  \partcombineAutomatic e2^"auto" e |
  \partcombineAutomatic c2^"auto" c |
  \partcombineAutomaticPar c2^"apart" \once \partcombineChords e^"chord once" |
  c2 c |
}

instrumentTwo = \relative {
  c'2 c |
  e2 e |
  a,2 c |
  c2 c' |
  c2 c |
  c2 c |
}
\end{verbatim}

<<
\new Staff { \instrumentOne }
\new Staff { \instrumentTwo }
\new Staff { \partcombine \instrumentOne \instrumentTwo }
>>

Ausgewählte Schnipsel

_Zwei Stimmen auf einem System kombinieren_

Die Funktion, die Stimmen kombiniert (also der \partcombine-Befehl) ermöglicht die Kombination unterschiedlicher Stimmen auf einem System. Textanweisungen wie "solo" oder "a2" werden automatisch hinzugefügt. Wenn man sie entfernen will, muss man die Eigenschaft printPartCombineTexts auf falsch setzen. Für Klavierauszüge muss natürlich kein "solo"/"a2" usw. hinzugefügt werden, man sollte sie also ausschalten. Wenn aber Solo-Stellen in einem Klavierauszug oder einer Chorpartitur angezeigt werden, ist es besser, normale Polyphonie zu
Kapitel 1: Musikalische Notation 169

verwenden, weil so die Solostellen angezeigt werden, auch wenn der Text des Stimmenkombinierers ausgeschaltet ist.

Der Schnipsel zeigt drei Möglichkeiten, Stimmen auf einem System zu kombinieren: Standardpolyphonie, `\partcombine` ohne Text und `\partcombine` mit Text.

```latex
%%% Combining pedal notes with clef changes

\begin{music}
\relative c'\{
\time 4/4
\begin{music}
a4 c4. (g8) a4 |
g4 e' g, (a8 b) |
c b a2.
\end{music}
\end{music}
\begin{music}
g4 e4. (d8) c4 |
r2 g'4 (f8 e) |
d2 \stemDown a
\end{music}
\end{music}

\score {
<<
\new Staff \with { instrumentName = "Standard polyphony" }
\new Staff \with { instrumentName = "PartCombine without text"
\printPartCombineTexts = ##f
}\partcombine \musicUp \musicDown
\new Staff \with { instrumentName = "PartCombine with text"
\partcombine \musicUp \musicDown
>>
\layout {
indent = 6.0\cm
\context {\Score
\override SystemStartBar.collapse-height = #30
}\}
}
```
Partcombine-Text ändern

Wenn Stimmen automatisch kombiniert werden, kann der Text, der für Solo- und Unisono-Stellen ausgegeben wird, geändert werden:
\new Staff <<
\set Staff.soloText = #"girl"
\set Staff.soloIIText = #"boy"
\set Staff.aDueText = #"together"
\partcombine
\relative c'' { g4 g r r a2 g }
\relative c'' { r4 r a( b) a2 g }
>>
girl boy together

Siehe auch

Musikglossar: Abschnitt “a due” in Glossar, Abschnitt “part” in Glossar.
Notationsreferenz: Abschnitt 1.6.3 [Orchesterstimmen erstellen], Seite 191.
Schnipsel: Abschnitt “Simultaneous notes” in Schnipsel.

Bekannte Probleme und Warnungen

Alle \partcombine...-Funktionen können nur zwei Stimmen aufnehmen und sind nicht für die Behandlung von Gesangstext geeignet. Das heißt, dass die Funktion nicht funktioniert, wenn einer der Kontexte explizit eine Bezeichnung erhält, um ihm etwa Gesangstext zuweisen zu können.

\partcombine...-Funktionen können nicht innerhalb von \timeses- oder \relative-Umgebungen geschrieben werden.

Wenn printPartCombineTexts definiert ist und die zwei Stimmen die gleichen Noten abwechselnd spielen, kann es sein, dass die Kombinationsfunktion den Text „a2“ mehr als einmal im gleichen Takt setzt.
\partcombine merkt nur, wenn eine Note in einer Voice-Umgebung beginnt; Information, dass eine Note in einer Stimme etwa schon begonnen hat, wenn die andere Stimme die gleiche Note spielt, fehlt. Das kann zu einigen unerwarteten Problemen führen, wie etwa dass „Solo“ bzw. „Unison“ falsch gesetzt werden.

\partcombine hält alle Strecker (Haltebögen, Legatobögen, Klammern usw.) in der gleichen Voice-Umgebung, sodass es sein kann, dass diese Strecker nicht korrekt erscheinen, wenn sie in einer anderen Stimme enden.

Wenn die \partcombine-Funktion beide Noten nicht kombinieren kann (beispielsweise weil beide Stimmen unterschiedliche Dauern haben), werden die Stimmen intern in one und two benannt. Das heißt, dass jeglicher Kontextwechsel zu einem Voice-Kontext mit anderer Bezeichnung ignoriert wird.

Lesen Sie auch den Abschnitt Bekannte Probleme und Warnungen, wenn Sie \partcombine in Tabulaturen benutzen (siehe [Standardtabulaturen], Seite 322).

**Musik parallel notieren**


\begin{quote}
\textbf{Achtung:} Taktüberprüfungen \textbackslash{}müssen benutzt werden, und die Takte müssen die gleiche Länge haben.
\end{quote}

\lparal1\parallelMusic voiceA,voiceB,voiceC {
  \% Bar 1
  r8 g'16 c'' e'' g' c'' e'' r8 g'16 c'' e'' g' c'' e'' |
  r16 e'8.~ 4 r16 e'8.~ 4 |
  c'2 c'2 |

  \% Bar 2
  r8 a'16 d'' f'' a' d'' f'' r8 a'16 d'' f'' a' d'' f'' |
  r16 d'8.~ 4 r16 d'8.~ 4 |
  c'2 c'2 |
}
\new StaffGroup <<
  \new Staff << \voiceA \.. \.. \voiceB >>
  \new Staff { \clef bass \voiceC }
\>>

Der relative Modus kann auch benutzt werden. Beachten Sie, dass der \relative-Befehl nicht innerhalb von \parallelMusic benutzt wird. Die Noten sind parallel zu der vorherigen Note
der gleichen Stimme, nicht zu der vorherigen Note in der Quelldatei. Anders gesagt ignorieren relative Noten von voiceA die Noten von voiceB.

\parallelMusic voiceA,voiceB,voiceC { 
\% Bar 1
r8 g16 c e g, c e r8 g,16 c e g, c e |
r16 e8.~ 4 r16 e8.~ 4 |
c2 c |

\% Bar 2
r8 a,16 d f a, d f r8 a,16 d f a, d f |
r16 d8.~ 4 r16 d8.~ 4 |
c2 c |
}
\new StaffGroup << 
\new Staff << \relative c'' \voiceA \relative c' \voiceB >> 
\new Staff \relative c' { \clef bass \voiceC }
>>

Das funktioniert ziemlich gut für Klaviernoten. Dieses Beispiel speichert vier konsekutive Takte in vier Variablen:

global = {
\key g \major
\time 2/4
}

\parallelMusic voiceA,voiceB,voiceC,voiceD { 
\% Bar 1
a8 b c d |
d4 e |
c16 d e fis d e fis g |
a4 a |

\% Bar 2
e8 fis g a |
fis4 g |
e16 fis g a fis g a b |
a4 a |

\% Bar 3 ...
}
\score {
\new PianoStaff <<
Kapitel 1: Musikalische Notation

\new Staff {
  \global
  <<
    \relative c'' \voiceA
    \ \n    \relative c' \voiceB
  >>
}
\new Staff {
  \global \clef bass
  <<
    \relative c \voiceC
    \ \n    \relative c \voiceD
  >>
}

Siehe auch


Schnipsel: Abschnitt “Simultaneous notes” in Schnipsel.

1.6 Notation auf Systemen
Dieser Abschnitt zeigt, wie die Erscheinung von Systemen beeinflusst wird, wie Partituren mit mehr als einem System gesetzt werden und wie man Aufführungsanweisungen und Stichnoten zu einzelnen Systemen hinzufügt.

### 1.6.1 Systeme anzeigen lassen

Dieser Abschnitt zeigt unterschiedliche Methoden, Notensysteme und Gruppen von Systemen zu erstellen.

#### Neue Notensysteme erstellen

Notensysteme (engl. *staff*, Pl. *staves*) werden mit dem \new oder \context-Befehl erstellt. Zu Einzelheiten siehe Abschnitt 5.1.2 [Kontexte erstellen und referenzieren], Seite 554.

Der einfachste Notensystem-Kontext ist **Staff**:

```latex
\new Staff \relative { c''4 d e f }
```


```latex
\new DrumStaff {
    \drummode { cymc hh ss tomh }
}
```

RhythmicStaff (Rhythmus-System) erstellt ein Notensystem mit nur einer Notenlinie, auf welcher nur die rhythmischen Werte der eingegebenen Noten dargestellt werden. Die wirklichen Längen bleiben erhalten. Zu Einzelheiten, siehe [Melodierhythmus anzeigen], Seite 75.

```latex
\new RhythmicStaff { c4 d e f }
```

TabStaff (Tabulatursystem) erstellt eine Tabulatur mit sechs Saiten in der üblichen Gitarrenstimmmung. Zu Einzelheiten siehe [Standardtabulaturen], Seite 322.

```latex
\new TabStaff \relative { c''4 d e f }
```
Es gibt zwei Notensysteme, die zur Notation von Alter Musik eingesetzt werden: MensuralStaff and VaticanaStaff. Sie sind erklärt in [Vordefinierte Umgebungen], Seite 412.

Das GregorianTranscriptionStaff (System zur Transkription des Gregorianischen Chorals) erstellt ein Notensystem, um modernen Gregorianischen Choral zu notieren. Es hat keine Notenlinien.

\new GregorianTranscriptionStaff \relative { c''4 d e f e d }

Neue Notensystem-Kontexte können selber definiert werden. Zu Einzelheiten, siehe Abschnitt 5.1.6 [Neue Kontexte definieren], Seite 565.

Siehe auch

Glossar: Abschnitt “staff” in Glossar, Abschnitt “staves” in Glossar.

Notationsreferenz: Abschnitt 5.1.2 [Kontexte erstellen und referenzieren], Seite 554, [Schlagzeugsysteme], Seite 367, [Melodierhythmus anzeigen], Seite 75, [Standardtabulaturen], Seite 322, [Vordefinierte Umgebungen], Seite 412, [Das Notensystem], Seite 181, [Gregorianische Gesangs-Kontexte], Seite 421, [Mensural-Kontexte], Seite 414, Abschnitt 5.1.6 [Neue Kontexte definieren], Seite 565.

Schnipsel: Abschnitt “Staff notation” in Schnipsel.

Referenz der Interna: Abschnitt “Staff” in Referenz der Interna, Abschnitt “DrumStaff” in Referenz der Interna, Abschnitt “GregorianTranscriptionStaff” in Referenz der Interna, Abschnitt “RhythmicStaff” in Referenz der Interna, Abschnitt “TabStaff” in Referenz der Interna, Abschnitt “MensuralStaff” in Referenz der Interna, Abschnitt “VaticanaStaff” in Referenz der Interna, Abschnitt “StaffSymbol” in Referenz der Interna.

Systeme gruppieren


Wenn kein Kontext angegeben ist, wird die Standardeinstellung eingesetzt: die Gruppe beginnt mit einer vertikalen Linie und die Taktlinien sind nicht verbunden.

<<
\new Staff \relative { c''1 c }
\new Staff \relative { c''1 c }
>>

Im StaffGroup-Kontext die Gruppe mit einer eckigen Klammer begonnen und die Taktlinien durch alle Systeme gezogen.

\new StaffGroup <<
Kapitel 1: Musikalische Notation

In einem **ChoirStaff** (Chorsystem) beginnt die Gruppe mit einer eckigen Klammer, aber die Taktlinien sind nicht verbunden.

```
\new ChoirStaff <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```

In einem **GrandStaff** (Akkolade) beginnt die Gruppe mit einer geschweiften Klammer und die Taktlinien sind durchgezogen.

```
\new GrandStaff <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```

Der **PianoStaff**-(Klaviersystem)-Kontext ist identisch mit dem **GrandStaff**-Kontext, aber es ermöglicht zusätzlich direkt die Angabe einer Instrumentbezeichnung. Zu Einzelheiten siehe [Instrumentenbezeichnungen], Seite 191.

```
\new PianoStaff <<
  \set PianoStaff.instrumentName = "Piano"
  \new Staff \relative { c''1 c }
  \new Staff \relative { \clef bass c1 c }
>>
```
Jede Systemgruppe stellt die Eigenschaft `systemStartDelimiter` (SystemBeginnBegrenzer) auf einen der folgenden Werte: `SystemStartBar`, `SystemStartBrace` oder `SystemStartBracket`. Ein viertes Begrenzer, `SystemStartSquare`, ist auch erreichbar, aber man muss ihr explizit einstellen.

Neue Systemgruppen können definiert werden. Zu Einzelheiten siehe Abschnitt 5.1.6 [Neue Kontexte definieren], Seite 565.

**Ausgewählte Schnipsel**

*Eine eckige Klammer zu Beginn von Systemgruppen benutzen*

Die Klammer zu Beginn von Systemgruppen kann auch in eine eckige Klammer (`SystemStartSquare`) umgewandelt werden, wenn man sie explizit im `StaffGroup`- oder `ChoirStaffGroup`-Kontext setzt.

```
\score { 
  \new StaffGroup { 
    \set StaffGroup.systemStartDelimiter = #'SystemStartSquare 
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  } 
}
```

*Klammer anzeigen, wenn nur ein System gesetzt wird*

Wenn nur ein System einer Systemgruppe vom Typ `ChoirStaff` oder `StaffGroup` angezeigt wird, wird die Klammer zu Beginn normalerweise nicht gesetzt. Das kann verändert werden, indem man die entsprechende Eigenschaft verändert.

Bei Systemen wie `PianoStaff` und `GrandStaff`, die mit einer geschweiften Klammer beginnen, muss eine andere Eigenschaft verändert werden, wie das zweite Beispiel zeigt.

```
\score { 
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = #4
    \override Score.SystemStartBar.collapse-height = #4
    \new Staff {
        c'1
    }
  >> 
}
\score { 
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = #4
```


\override Score.SystemStartBar-collapse-height = #4
\new Staff {
  \clef treble
  c'1
}
}  
}\new Staff {
  \clef treble  
  c'1
}
}  
\new StaffGroup \relative c'' {
<<  
  \new Staff { << \global { c1 c } >> }  
  \new Staff { << \global { c c } >> }  
}  
\new StaffGroup \relative c'' {
<<  
  \new Staff { << \global { c1 c } >> }  
  \new Staff { << \global { c c } >> }  
}  

\new StaffGroup \relative c'' {
<<  
  \new Staff { << \global { c1 c } >> }  
  \new Staff { << \global { c c } >> }  
}  

\new StaffGroup \relative c'' {
<<  
  \new Staff { << \global { c1 c } >> }  
  \new Staff { << \global { c c } >> }  
}  

Siehe auch

Glossar: Abschnitt “brace” in Glossar, Abschnitt “bracket” in Glossar, Abschnitt “grand staff” in Glossar.

Notationsreferenz: [Instrumentenbezeichnungen], Seite 191, Abschnitt 5.1.6 [Neue Kontexte definieren], Seite 565.

Schnipsel: Abschnitt “Staff notation” in Schnipsel.

Referenz der Interna: Abschnitt “Staff” in Referenz der Interna, Abschnitt “StaffGroup” in Referenz der Interna, Abschnitt “ChoirStaff” in Referenz der Interna, Abschnitt “GrandStaff” in

Bekannte Probleme und Warnungen

PianoStaff nimmt standardmäßig keine ChordNames (Akkordbezeichnungen) auf.

Verschachtelte Notensysteme

System-Gruppen können in beliebiger Tiefe geschachtelt werden. In diesem Fall erstellt jeder neue, innen liegende Kontext eine neue Klammer außerhalb der Klammer der Systemgruppe, in der er sich befindet.

```
\new StaffGroup <<
  \new Staff \relative { c''2 c | c2 c }
\new StaffGroup <<
  \new Staff \relative { g'2 g | g2 g }
  \new StaffGroup \with {
    systemStartDelimiter = #'SystemStartSquare
  }
  \new Staff \relative { e'2 e | e2 e }
  \new Staff \relative { c'2 c | c2 c }
>>
>>
```

Neue geschachtelte Systemgruppen können definiert werden. Zu Einzelheiten siehe Abschnitt 5.1.6 [Neue Kontexte definieren], Seite 565.

Ausgewählte Schnipsel

*Systeme schachteln*

Die Eigenschaft `systemStartDelimiterHierarchy` kann eingesetzt werden, um komplizierte geschachtelte Systemklammern zu erstellen. Der Befehl `\set StaffGroup.systemStartDelimiterHierarchy` nimmt eine Liste mit der Anzahl der Systeme, die ausgegeben werden, auf. Vor jedem System kann eine Systemanfangsklammer angegeben werden. Sie muss in Klammern eingefügt werden und umfasst so viele Systeme,

\new StaffGroup
\relative c' <<
  \set StaffGroup.systemStartDelimiterHierarchy
  = #'(SystemStartSquare (SystemStartBrace (SystemStartBracket a
          (SystemStartSquare b) ) c ) d)
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
\new Staff { c1 }
>>

Siehe auch

Notationsreferenz: [Systeme gruppieren], Seite 175, [Instrumentenbezeichnungen], Seite 191, Abschnitt 5.1.6 [Neue Kontexte definieren], Seite 565.

Schnipsel: Abschnitt “Staff notation” in Schnipsel.


Systeme trennen

Wenn die Anzahl der Systeme sich von Seite zu Seite ändert, wird normalerweise ein Trennzeichen hinzugefügt, dass die Systeme voneinander trennt. Die Standardeinstellung ist, dass der Trenner nicht gesetzt wird, aber man kann ihn mit einer Option in der \paper-Umgebung angeschalten.

\book {
  \score {
    \new StaffGroup <<
    \new Staff {

\
Kapitel 1: Musikalische Notation

\relative {
   \relative { 
      c''4 c c c
      \break
      c4 c c c
   } 
}
\new Staff { 
   \relative { 
      c''4 c c c
      \break
      c4 c c c
   } 
} 
>> 
\paper {
   system-separator-markup = \slashSeparator 
   % following commands are needed only to format this documentation
   paper-width = 100\mm
   paper-height = 100\mm
   tagline = ##f
} 

\begin{music}
   \begin{staff}
      \relative { 
         c''4 c c c 
         \break 
         c4 c c c 
      } 
   \end{staff}
   \begin{staff}
      \relative { 
         c''4 c c c 
         \break 
         c4 c c c 
      } 
   \end{staff}
\end{music}

Siehe auch
Notationsreferenz: Abschnitt 4.1 [Seitenlayout], Seite 498.
Schnipsel: Abschnitt “Staff notation” in Schnipsel.

1.6.2 Einzelne Systeme verändern

Das Notensystem
Die Befehle \stopStaff und \startStaff können benutzt werden, um ein Notensystem zu stoppen oder (wieder) zu beginnen.
Vordefinierte Befehle

\startStaff, \stopStaff. Die Linien eines Notensystems gehören zu dem StaffSymbol-(NotensystemSymbol)-Grob (hierzu gehören auch Hilfslinien). StaffSymbol-Eigenschaften können verändert werden, um die Erscheinung des Notensystems zu beeinflussen, aber sie müssen gesetzt werden, bevor das System erstellt wird.

Die Anzahl der Notenlinien kann verändert werden:
\relative {
  \stopStaff f''4 d \startStaff g, e
  f'4 d \stopStaff g, e
  f'4 d \startStaff g, e
}\n
\relative {
  f''4 d \stopStaff
  \override Staff.StaffSymbol.line-count = #2
  \startStaff g, e |

  f'4 d \stopStaff
  \revert Staff.StaffSymbol.line-count
  \startStaff g, e |
}\n
die Position des Notenschlüssels und die Position von c' können geändert werden, um dem neuen System zu entsprechen. Siehe auch [Notenschlüssel], Seite 16.

\new Staff \with {
    \override StaffSymbol.thickness = #3
} \relative {
    f''4 d g, e
}


\new Staff \with {
    \override StaffSymbol.thickness = #2
    \override StaffSymbol.ledger-line-thickness = #'(0.5 . 0.4)
} \relative {
    f''''4 a, a,, f
}

Die vertikale Position der Hilfslinien kann verändert werden:

\new Staff \with {
    \override StaffSymbol.ledger-positions = #'(-3 -2 -1 2 5 6)
} \relative {
    f''''4 a, a,, f
}

Zusätzliche Hilfslinien können auch innerhalb des Systems gesetzt werden, wenn sie von einem selbstdifinierten Notensystem genötigt werden. Das Beispiel zeigt die Standardposition der Hilfslinien wenn die explizite Position mit legder-position nicht definiert ist. Der Befehl stopStaff wird benötigt, damit der Befehl sich auf das gesamte System (StaffSymbol) auswirkt.
Der Abstand zwischen Notenlinien kann verändert werden. Diese Einstellung wirkt sich auch auf den Abstand der Hilfslinien aus.

\new Staff \with {
  \override StaffSymbol.staff-space = #1.5
} \relative {
  \override Staff.StaffSymbol.line-positions = #'(-4 -2 -0.2 0 0.2 2 4)
  \override StaffSymbol.staff-space = #1.5
  f'''4 d, g, e,
}

Ausgewählte Schnipsel

Eine Linie des Notensystems dicker als die anderen machen

Für den pädagogischen Einsatz kann eine Linie des Notensystems dicker gezeichnet werden (z. B. die Mittellinie, oder um den Schlüssel hervorzuheben). Das ist möglich, indem man zusätzliche Linien sehr nahe an der Linie, die dicker erscheinen soll, einfügt. Dazu wird die line-positions-Eigenschaft herangezogen.

Siehe auch

Glossar: Abschnitt “line” in Glossar, Abschnitt “ledger line” in Glossar, Abschnitt “staff” in Glossar.

Notationsreferenz: [Notenschlüssel], Seite 16.
Schnipsel: Abschnitt “Staff notation” in Schnipsel.
Referenz der Interna: Abschnitt “StaffSymbol” in Referenz der Interna, Abschnitt “staff-symbol-interface” in Referenz der Interna.

Ossia-Systeme

Ossia-Systeme können gesetzt werden, indem zwei gleichzeitige Notensysteme an der entsprechenden Position erstellt werden:

\new Staff \relative {
  c'4 b d c
  <<
    \new Staff { c4 b d c }
  \new Staff { e4 d f e }
  >>
}
Dieses Beispiel ist aber normalerweise nicht erwünscht. Um Ossia-Systeme zu setzen, die sich über dem eigentlichen System befinden, keine Takt- und Schlüsselangaben haben und kleiner gesetzt sind, müssen einige Optimierungen angewendet werden. Im Handbuch zum Lernen wird eine Technik vorgestellt, mit der das gewünschte Ergebnis erreicht werden kann, beginnend in Abschnitt “Musikalische Ausdrücke ineinander verschachteln” in Handbuch zum Lernen.

Das Beispiel unten setzt die alignAboveContext-(oberhalbAusrichtenKontext)-Eigenschaft ein, um den Ossia-Abschnitt auszurichten. Diese Methode bietet sich an, wenn nur einige Ossia-Systeme benötigt werden.

```latex
\new Staff = main \relative { 
\new Staff \with { 
  \remove "Time_signature_engraver"
  \alignAboveContext = "main"
  \fontSize = #-3
  \override StaffSymbol.staff-space = #(magstep -3)
  \override StaffSymbol.thickness = #(magstep -3)
  \firstClef = ##f
} 
{ c'\'4 b d c }
<<
\new Staff \with { 
  \remove "Time_signature_engraver"
  \alignAboveContext = "main"
  \fontSize = #-3
  \override StaffSymbol.staff-space = #(magstep -3)
  \override StaffSymbol.thickness = #(magstep -3)
  \firstClef = ##f
} 
{ e4 d f e }
>>
c4 b c2
}
```

Wenn mehrere isolierte Ossia-Systeme gebraucht werden, kann es günstiger sein, einen leeren Staff-Kontext mit einer spezifischen Kontextidentifikation zu erstellen. Die Ossia-Abschnitte werden dann erstellt, indem dieser Kontext aufgerufen wird und mit `\startStaff` und `\stopStaff` an den richtigen Stellen sichtbar gemacht wird. Der Vorteil dieser Methode zeigt sich, wenn man längere Stücke setzt.

```latex
<<
\new Staff = ossia \with { 
  \remove "Time_signature_engraver"
  \hide Clef
```

\new Staff = ossia \with { 
  \remove "Time_signature_engraver"
  \hide Clef
```
Kapitel 1: Musikalische Notation

Man kann auch den \Staff \RemoveEmptyStaves-Befehl einsetzen, um Ossia-Systeme zu erstellen. Diese Methode eignet sich am besten, wenn nach dem Ossia sofort ein Zeilenumbruch erfolgt. Mehr Information zu \Staff \RemoveEmptyStaves findet sich in [Systeme verstecken], Seite 188.
Ausgewählte Schnipsel

_Gesangstext und Ossia vertikal ausrichten_

Dieser Schnipsel zeigt, wie man die Kontexteigenschaften `alignBelowContext` und `alignAboveContext` benutzen kann, um die Positionierung von Gesangstext und Ossia-Abschnitten zu kontrollieren.

\paper {
  ragged-right = ##t
}

\relative c' <<
\new Staff = "1" { c4 c s2 } \new Staff = "2" { c4 c s2 } \new Staff = "3" { c4 c s2 }
{ \skip 2 <<
\lyrics {
  \set alignBelowContext = #"1"
  lyrics4 below
}
}
Siehe auch

Glossar: Abschnitt “ossia” in Glossar, Abschnitt “staff” in Glossar, Abschnitt “Frenched staff” in Glossar.


Notationsreferenz: [Systeme verstecken], Seite 188.

Schnipsel: Abschnitt “Staff notation” in Schnipsel.

Referenz der Interna: Abschnitt “StaffSymbol” in Referenz der Interna.

Systeme verstecken

Die Notenlinien können entfernt werden, indem der Staff_symbol_engraver aus dem Staff-Kontext entfernt wird. Alternativ kann auch \stopStaff eingesetzt werden.

\new Staff \with {
    \remove "Staff_symbol_engraver"
}
Leere Systeme können versteckt werden, wenn der \Staff \RemoveEmptyStaves-Befehl im \layout-Abschnitt benutzt wird. In großen Orchesterpartituren wird dies oft verwendet, um die leeren Systeme von gerade nicht spielenden Instrumenten zu verstecken. In der Standardeinstellung werden alle leeren Notenzeilen außer die des ersten Systems entfernt.

\begin{Verbatim}
\layout {
  \context {
    \Staff \RemoveEmptyStaves
  }
}
\end{Verbatim}

\relative <<
  \new Staff {
    e'4 f g a \break
    b1 \break
    a4 b c2
  }
  \new Staff {
    c,4 d e f \break
    R1 \break
    f4 g c,2
  }
\end{Verbatim}

\Staff \RemoveEmptyStaves kann auch eingesetzt werden, um Ossiaabschnitte zu erstellen. Zu Einzelheiten, siehe [Ossia-Systeme], Seite 184.

Der \VaticanaStaff \RemoveEmptyStaves-Befehl kann benutzt werden, um leere Takte in Notation der Alten Musik zu entfernen. Gleichermaßen kann \RhythmicStaff
\RemoveEmptyStaves eingesetzt werden, um leere Takte in einem RhythmicStaff-Kontext zu entfernen.

Vordefinierte Befehle
\Staff \RemoveEmptyStaves, \VaticanaStaff \RemoveEmptyStaves, \RhythmicStaff \RemoveEmptyStaves.

Ausgewählte Schnipsel

Die erste leere Notenzeile auch entfernen


Das untere Notensystem der zweiten Systemgruppe wird nicht entfernt, weil in die Einstellungen in dem Schnipsel nur für das eine Notensystem gültig sind.

\layout {
  \context {
  \Staff \RemoveEmptyStaves
  \% To use the setting globally, uncomment the following line:
  \% \override VerticalAxisGroup.remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
c1
  }
  \new Staff {
    \% To use the setting globally, comment this line,
    \% uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
>>
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
c1
  }
  \new Staff {
    \% To use the setting globally, comment this line,
    \% uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
>>
Siehe auch

Glossar: Abschnitt “Frenched staff” in Glossar.


Notationsreferenz: Abschnitt 5.1.5 [Die Standardeinstellungen von Kontexten ändern], Seite 560, [Das Notensystem], Seite 181, [Ossia-Systeme], Seite 184, [Unsichtbare Noten], Seite 207, Abschnitt 5.4.6 [Sichtbarkeit von Objekten], Seite 589.

Schnipsel: Abschnitt “Staff notation” in Schnipsel.


Bekannte Probleme und Warnungen

Wenn man den Staff_symbol_engraver entfernt, werden auch die Taktlinien entfernt. Wenn eine sichtbare Taktlinie angefordert wird, kann es zu Formatierungsfehlern kommen. In diesem Fall sollten folgende Befehle eingesetzt werden, anstatt den Engraver zu entfernen:

\omit StaffSymbol
\override NoteHead.no-ledgers = ##t

Zu den bekannten Fehlern und Warnungen, die mit \Staff \RemoveEmptyStaves zusammenhängen, siehe Abschnitt 5.1.5 [Die Standardeinstellungen von Kontexten ändern], Seite 560.

1.6.3 Orchesterstimmen erstellen

Dieser Abschnitt zeigt, wie man Tempo-Anweisungen und Instrumentenbezeichnungen eingibt. Es werden auch Möglichkeiten vorgestellt, andere Stimmen zu zitieren und Stichnoten zu formatieren.

Instrumentenbezeichnungen

Instrumentenbezeichnungen können an der linken Seite von Notensystemen im Staff-, PianoStaff-, StaffGroup, GrandStaff und ChoirStaff-Kontext gesetzt werden. Der Wert von instrumentName wird für das erste System eingesetzt, der Wert von shortInstrumentName für alle weiteren Systeme.

\new Staff \with {
instrumentName = "Violin"
shortInstrumentName = "Vln."
}
Mit dem Textbeschriftungsmodus (\markup) können auch komplizierte Instrumentenbezeichnungen erstellt werden:

\new Staff \with {
  instrumentName = \markup {
    \column { "Clarinetti"
      \line { "in B" \smaller \flat }
    }
  }
} \relative {
  c''4 c,16 d e f g2
}

Wenn zwei oder mehr Systeme gruppiert werden, werden die Instrumentenbezeichnungen automatisch zentriert. Um auch mehrzeilige Instrumentenbezeichnungen zentriert zu setzen, muss \center-column benutzt werden:

<<
\new Staff \with {
  instrumentName = "Flute"
}
{ f2 g4 f }
\new Staff \with {
  instrumentName = \markup {
    \center-column { "Clarinet" }
    \line { "in B" \smaller \flat }
  }
}
{ c4 b c2 }
>>
WENN die Instrumentenbezeichnung zu lang ist, kann es vorkommen, dass die Bezeichnungen in einer Gruppe nicht zentriert werden. Um dennoch eine Zentrierung zu erhalten, müssen die Werte des Einzugs (\texttt{indent} und \texttt{short-indent}) vergrößert werden. Zu Einzelheiten siehe \cite{paper-Variablen für Verschiebungen und Einrückungen}, Seite 505.

\begin{verbatim}
<< 
\new Staff \with {
  \relative {
    f''2 g4 f \break
    g4 f g2
  }
} 
\new Staff \with {
  \relative {
    c''4 b c2 \break
    c2 b4 c
  }
} 
>> 
\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm
}
\new Staff \with {
  instrumentName = "Alto Flute in G"
  shortInstrumentName = "Flt."
} 
\new Staff \with {
  instrumentName = "Clarinet"
  shortInstrumentName = "Clar."
} 
\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm
}
\end{verbatim}

Um Instrumentenbezeichnungen zu anderen Kontexten (wie etwa \texttt{ChordNames} or \texttt{FiguredBass}) hinzuzufügen, muss der \texttt{Instrument_name_engraver} dem entsprechenden Kontext hinzugefügt werden. Zu Einzelheiten siehe Abschnitt 5.1.4 [Umgebungs-Plugins verändern], Seite 558.

Die kurzen Instrumentenbezeichnungen können mitten in einer Partitur geändert werden. Die Lange Bezeichnung \texttt{instrumentName} wird nur bei ihrem ersten Auftreten gesetzt und spätere Änderungen nicht berücksichtigt.

\begin{verbatim}
\new Staff \with {
  instrumentName = "Flute"
  shortInstrumentName = "Flt."
}
\end{verbatim}
Wenn das Instrument gewechselt werden soll, kann der Befehl \addInstrumentDefinition in Begleitung von \instrumentSwitch benutzt werden, um eine detaillierte Auflistung aller notwendigen Änderungen für den Wechsel zu definieren. Der \addInstrumentDefinition-Befehl hat zwei Argumente: eine Identifikation und eine Assoziationsliste von Kontexteigenschaften und Werten, die für dieses Instrument benutzt werden müssen. Der Befehl muss sich auf der höchsten Ebene in der Eingabedatei befinden. \instrumentSwitch wird dann benutzt, um den Wechsel vorzunehmen:

\addInstrumentDefinition "contrabassoon"
#`((instrumentTransposition . ,(ly:make-pitch -1 0 0))
  (shortInstrumentName . "Cbsn.")
  (clefGlyph . "clefs.F")
  (middleCPosition . 6)
  (clefPosition . 2)
  (instrumentCueName . ,(make-bold-markup "cbsn.")
  (midiInstrument . "bassoon"))

\new Staff \with {
  instrumentName = "Bassoon"
}
\relative c' {
  \clef tenor
  \compressFullBarRests
  c2 g'
  R1*16
Siehe auch

Notationsreferenz: \[paper-Variablen für Verschiebungen und Einrückungen\], Seite 505, Abschnitt 5.1.4 [Umgebungs-Plugins verändern], Seite 558.

Schnipsel: Abschnitt “Staff notation” in Schnipsel.

Referenz der Interna: Abschnitt “InstrumentName” in Referenz der Interna, Abschnitt “PianoStaff” in Referenz der Interna, Abschnitt “Staff” in Referenz der Interna.

Andere Stimmen zitieren
Es kommt sehr oft vor, dass eine Orchesterstimme die gleichen Noten wie eine andere spielt. So können etwa die ersten und zweiten Geigen für eine bestimmte Passage die gleichen Noten haben. In LilyPond kann man das erreichen, indem eine Stimme von der anderen zitiert, sodass man die Noten für die zweite Stimme nicht noch einmal eingeben muss.

Der \addQuote-Befehl, auf höchster Ebene in der Datei, definiert einen Notenabschnitt, aus dem zitiert werden kann.

Der \quoteDuring-Befehl wird benutzt, um den Punkt anzuzeigen, an dem das Zitat beginnt. Er benötigt zwei Argumente: die Bezeichnung der zitierten Stimme, wie vorher mit \addQuote definiert, und einen musikalischen Ausdruck für die Dauer des Zitates.

\begin{Verbatim}
fluteNotes = \relative {
  a'4 gis g gis | b4"quoted" r8 ais\p a4( f)
}

oboeNotes = \relative {
  c''4 cis c b \quoteDuring "flute" { s1 }
}
\addQuote "flute" { \fluteNotes }
\end{Verbatim}

\begin{Verbatim}
\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
\end{Verbatim}
Wenn der musikalische Ausdruck, der mit dem \quoteDuring-Befehl benutzt wird, antel von unsichtbare Noten oder Ganztaktpausen etwa Noten enthält, wird eine polyphone Stelle begonnen, was meistens nicht erwünscht ist:

```latex
\texttt{\textbackslash fluteNotes = \relative { \\
            a'4 gis g gis | b4"quoted" r8 ais\p a4( f) \\
}\textbackslash oboeNotes = \relative { \\
            c''4 cis c b \quoteDuring "flute" { e4 r8 ais b4 a } \\
}\texttt{\\addQuote "flute" { \fluteNotes } \\
\\score { \\
    \<< \\
    \new Staff \with { instrumentName = "Flute" } \fluteNotes \\
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes \\
    \>> \\
}\texttt{\textbackslash clarinetNotes = \relative c'' { \\
            \transposition bes \key d \major \\
            b4 ais a ais | cis4"quoted" r8 bis\p b4( f) } \\
}\texttt{\textbackslash oboeNotes = \relative { \\
            c''4 cis c b \quoteDuring "clarinet" { s1 } \\
}\texttt{\\addQuote "clarinet" { \clarinetNotes } \\
\\score { \\
}
```

Der \quoteDuring-Befehl benutzt die Einstellungen des \transposition-Befehls beider Stimmen, um Noten für die zitiierende Stimme zu produzieren, die das gleiche klingende C wie die zitierte Stimme haben.

```latex
fluteNotes = \relative {
  a'2 g2 |
  b4\"quoted\" r8 ais a4\f( c->)
}

oboeNotes = \relative {
  c''2. b4 |
  \quoteDuring "flute" { s1 }
}

\addQuote "flute" { \fluteNotes }
\score {
  \set Score.quotedEventTypes = #'(note-event articulation-event
crescendo-event rest-event
slur-event dynamic-event)
  \new Staff \with { instrumentName = "Flute" } \fluteNotes
  \new Staff \with { instrumentName = "Oboe" } \oboeNotes
}
```

Zitate können auch mit Marken (engl. tag) versehen werden. Siehe auch [Marken benutzen], Seite 479.
Kapitel 1: Musikalische Notation

Siehe auch
Notationsreferenz: [Transposition von Instrumenten], Seite 23, [Marken benutzen], Seite 479.
Installierte Dateien: scm/define-event-classes.scm.
Schnitt: Abschnitt “Staff notation” in Schniptel.

Bekannte Probleme und Warnungen

Nur der Inhalt der ersten Stimme innerhalb eines \addQuote-Befehls wird für das Zitat herangezogen. Wenn der zitierte Ausdruck also \new oder \context Voice-Befehle enthält, werden deren Inhalte nicht zitiert. Zitieren von Ziernoten und Vorschläge wird von LilyPond nicht unterstützt und kann dazu führen, dass LilyPond abstürzt. Wenn geschachtelte Triolen zitiert werden, ist das Notenbild unter Umständen sehr schlecht.

Stichnoten formatieren

Die einfachste Art, Stichnoten zu erstellen, ist es, einen CueVoice-Kontext in der Stimme zu erstellen.
\relative {
  R1 <<
  \{ e'2\rest r4. e8 \}
  \new CueVoice {
    \stemUp d'8"flute" c d e fis2
  }
  >>
  d,,4 r a r
}

Der Befehl \cueClef kann auch in einem expliziten CueVoice-Kontext eingesetzt werden, wenn ein Schlüsselwechsel nötig ist. Hiermit wird ein Schlüssel entsprechender Größe für die Stichnoten ausgegeben. Der Befehl \cueClefUnset kann dann eingesetzt werden, um wieder zum ursprünglichen Schlüssel zurückzukehren, wiederum in der richtigen Größe:
\relative {
  \clef "bass"
  R1 <<
  \{ e'2\rest r4. \cueClefUnset e,8 \}
  \new CueVoice {
    \cueClef "treble" \stemUp d'8"flute" c d e fis2
  }
  >>
  d,,4 r a r
}
Die Befehle \texttt{\cueClef} und \texttt{\cueClefUnset} können auch ohne eine \texttt{CueVoice}-Umgebung eingesetzt werden:

\begin{verbatim}
relative {
  \clef "bass"
  R1
  \cueClef "treble"
  d''8""flute" c d e fis2
  \cueClefUnset
  d,,4 r a r
}
\end{verbatim}

Für kompliziertere Stichnotenbehandlung, etwa mit Transposition, oder um Stichnoten aus unterschiedlichen Stimmen einzufügen, können die Befehle \texttt{\cueDuring} oder \texttt{\cueDuringWithClef} eingesetzt werden. Sie stellen eine spezielle Form von \texttt{\quoteDuring} dar, siehe auch [Andere Stimmen zitieren], Seite 195.

Die Syntax lautet:

\begin{verbatim}
\cueDuring \#Zitatbezeichnung \#Richtung Noten
\end{verbatim}

sowie

\begin{verbatim}
\cueDuringWithClef \#Zitatbezeichnung \#Richtung \#Schlüssel \#Noten
\end{verbatim}

Die Noten der entsprechenden Takten von \texttt{Zitatbezeichnung} wird dem \texttt{CueVoice}-Kontext hinzugefügt und erscheint gleichzeitig mit \texttt{Noten}, wodurch eine polyphone Situation entsteht. Die \texttt{Richtung} kann entweder \texttt{UP} oder \texttt{DOWN} sein, womit die zitierten Noten entweder als erste oder als zweite Stimme in einem System gesetzt werden.

\begin{verbatim}
fluteNotes = \relative {
  \r2. \c''4 | \d8 \c \d \e fis2 | \g2 \d |
}

oboeNotes = \relative c'' {
  \R1 \new \CueVoice { \set instrumentCueName = "flute" }
  \cueDuring "flute" \#UP { \R1 }
    \g2 \c, 
}

\addQuote "flute" { \fluteNotes }
\new Staff {
  \oboeNotes
}
\end{verbatim}

Es ist möglich anzupassen, welche Objekte der Notation von \texttt{\cueDuring} zitiert werden, indem man die \texttt{quotedCueEventTypes}-Eigenschaft verändert. Ihr Standardwert ist
Kapitel 1: Musikalische Notation

'(note-event rest-event tie-event beam-event +tuplet-span-event); somit werden also nur Noten, Pausen, Bindebögen, Balken und N-tolen zitiert, nicht aber Artikulationen, Dynamik, Beschriftung usw.

Achtung: Wenn eine Voice-Umgebung mit \cueDuring beginnt, wie im folgenden Beispiel, muss die Voice-Umgebung explizit erstellt werden, weil sonst der gesamte musikalische Ausdruck zum CueVoice-Kontext gehören würde.

\begin{verbatim}
\begin{Verbatim}
\set Score.quotedCueEventTypes = #'(note-event rest-event tie-event beam-event tuplet-span-event
dynamic-event slur-event)
\cueDuring "oboe" #UP { \addQuote \"oboe\" { \oboeNotes }
\new Voice \relative c'' {
\set instrumentCueName = "flute"
\new Staff {
\fluteNotes = \relative {
 \r2. c''4 d8 c d e fis2 g2 d2
}
\bassoonNotes = \relative c {
 \clef bass
 R1
 \clef treble
 \new CueVoice { \set instrumentCueName = "flute" }
 \cueDuring "flute" #UP { R1 }
 \clef bass
 g4. b8 d2
}
\addQuote "flute" { \fluteNotes }
\end{Verbatim}
\end{verbatim}

Die Bezeichnung des gerade spielenden Instruments in den Stichnoten kann gesetzt werden, indem man die instrumentCueName-Eigenschaft in einen temporären CueVoice-Kontext setzt. Die Platzierung und der Stil von instrumentCueName wird durch das \instrumentSwitch-Objekt kontrolliert, siehe [Instrumentenbezeichnungen], Seite 191. Wenn die Stichnoten einen Schlüsselwechsel erfordern, kann dieser manuell hervorgerufen werden, aber der originale Schlüssel muss auch manuell am Ende der Stichnoten wieder hergestellt werden.
Alternativ kann auch die Funktion \cueDuringWithClef eingesetzt werden. Dieser Befehl erhält zusätzlich ein Argument, das den Schlüsselwechsel anzeigt, den man für die Stichnoten braucht. Der originale Schlüssel wird automatisch wieder hergestellt.

\begin{verbatim}
fluteNotes = \relative {
 r2. c'"4 d8 c d e fis2 g2 d2 }
\end{verbatim}

\begin{verbatim}
bassoonNotes = \relative c {
 \clef bass
 R1
 \new CueVoice { \set instrumentCueName = "flute" } \cueDuringWithClef "flute" #UP "treble" { R1 } g4. b8 d2 }
\end{verbatim}

\begin{verbatim}
\addQuote "flute" { \fluteNotes }
\new Staff {
 \bassoonNotes }
\end{verbatim}

Genauso wie \quoteDuring berücksichtigt auch \cueDuring Transpositionen. Stichnoten werden auf den Tonhöhen erstellt, die für das Instrument geschrieben würden, in dessen Noten sie gesetzt werden, um die klingenden Töne des Quelleninstruments zu produzieren.

Um Stichnoten anders zu transponieren, muss \transposedCueDuring benutzt werden. Dieser Befehl braucht ein zusätzliches Argument, um (in absolutem Modus) die gedruckte Tonhöhe vorzugeben, mit der das eingestrichene C dargestellt werden soll. Das ist nützlich, wenn man Stichnoten von einem Instrument mit einem vollständig anderen Register benutzt:

\begin{verbatim}
piccoloNotes = \relative {
 \clef "treble"8
 R1
 c'"8 c c e g2
c4 g g2 }
\end{verbatim}

\begin{verbatim}
bassClarinetNotes = \relative c' {
 \key d \major
 \transposition bes,
d4 r a r \transposedCueDuring "piccolo" #UP d { R1 }
d4 r a r
\end{verbatim}
Kapitel 1: Musikalische Notation

Der \killCues-Befehl entfernt Stichnoten aus einem musikalischen Ausdruck, sodass derselbe musikalische Ausdruck für die Partitur und für eine Stimme mit Stichnoten eingesetzt werden kann. Der Befehl \killCues entfernt nur Noten und Ereignisse, die durch \cueDuring zitiert wurden. Andere Beschriftungen in Verbindung mit Stichnoten, wie etwa Schlüsselwechsel und Marken, die das Ursprungsinstrument anzeigen, können mit Marken versehen werden, um sie selektiv einzufügen, siehe [Marken benutzen], Seite 479.

```latex
\definemacro{fluteNotes}{\relative {r2. c''4 d8 c d e fis2 g2 d2}}
\definemacro{bassoonNotes}{\relative c {\clef bass R1 \tag #'part {\clef treble \new CueVoice {\set instrumentCueName = "flute"}} \cueDuring "flute" #UP {R1 \tag #'part \clef bass g4. b8 d2}}}}
\addQuote "flute" {\fluteNotes}
\new Staff {\fluteNotes}
\new StaffGroup <<
\new Staff {\fluteNotes}
\new Staff {\removeWithTag #'part {\killCues {\bassoonNotes}}}
\nonBreakingSpace}\nonBreakingSpace=\nonBreakingSpace10x9x11=\nonBreakingSpace202}
```
Alternativ können Schlüsselwechsel und Instrumentenbezeichnungen auch in eine Instrument-Definition unter Einsatz des Befehls \addInstrumentDefinition zusammengefasst werden, siehe [Instrumentenbezeichnungen], Seite 191.

**Siehe auch**

Notationsreferenz: [Andere Stimmen zitieren], Seite 195, [Transposition von Instrumenten], Seite 23, [Instrumentenbezeichnungen], Seite 191, [Marken benutzen], Seite 479, [Notenschlüssel], Seite 16, [Musikalische Stichnoten], Seite 285.

Schnipsel: Abschnitt “Staff notation” in Schnipsel.


**Bekannte Probleme und Warnungen**

Zusammenstöße können bei Benutzung von \cueDuring zwischen Pausen der Hauptstimme und den Stichnoten des CueVoice-Kontexts auftreten. Wenn \cueDuringWithClef oder \transposedCueDuring eingesetzt wird, muss das zusätzliche Argument nach dem Zitat und der Richtung erscheinen.

### 1.7 Anmerkungen

Dieser Abschnitt zeigt die verschiedenen Möglichkeiten, die Erscheinung der Noten zu ändern und analytische bzw. pädagogische Anmerkungen anzubringen.

#### 1.7.1 Innerhalb des Systems

Dieser Abschnitt zeigt, wie man Elemente hervorhebt, die sich innerhalb des Notensystems befinden.
Auswahl der Notations-Schriftgröße


Achtung: Für Schriftgröße von Text, siehe [Überblick über die wichtigsten Textbeschriftungsbefehle], Seite 225.

\huge c4.-> d8---3
\large c4.-> d8---3
\normalsize c4.-> d8---3
\small c4.-> d8---3
\tiny c4.-> d8---3
\teeny c4.-> d8---3

Intern wird hiermit die \texttt{fontSize}-Eigenschaft gesetzt. Sie wird für alle Layout-Objekte definiert. Der Wert von \texttt{font-size} ist eine Zahl, die die Größe relativ zur Standardgröße für die aktuelle Systemhöhe angibt. Jeder Vergrößerungsschritt bedeutet etwa eine Vergrößerung um 12% der Schriftgröße. Mit sechs Schritten wird die Schriftgröße exakt verdoppelt. Die Scheme-Funktion \texttt{magstep} wandelt einen Wert von \texttt{font-size} in einen Skalierungsfaktor um. Die \texttt{font-size}-Eigenschaft kann auch direkt gesetzt werden, so dass sie sich nur auf bestimmte Layoutobjekte bezieht.

\set fontSize = #3
c4.-> d8---3
\override NoteHead.font-size = #-4
c4.-> d8---3
\override Script.font-size = #2
c4.-> d8---3
\override Stem.font-size = #-5
c4.-> d8---3


Die \texttt{font-size}-Eigenschaft kann nur für die Layoutobjekte gesetzt werden, die Schrift-Dateien benutzen. Das sind die, welche die \texttt{font-interface}-Layoutschnittstelle unterstützen.
Vordefinierte Befehle
\teeny, \tiny, \small, \normalsize, \large, \huge.

Siehe auch
Referenz der Interna: Abschnitt “font-interface” in Referenz der Interna.

Fingersatzanweisungen
Fingersatzanweisungen können folgenderweise notiert werden: ‘Note’-Zahl
\relative { c'4-1 d-2 f-4 e-3 }

Für Fingerwechsel muss eine Textbeschriftung (markup) benutzt werden:
c4-1 d-2 f-4 c\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.
\relative { <a'_\thumb a'-3>2 <b_\thumb b'-3> }

Fingersätze für Akkorde können auch zu einzelnen Noten hinzugefügt werden, indem sie innerhalb der Akkord-Klammer direkt an die Noten angefügt werden.
\relative {
  <c''-1 e-2 g-3 b-5>2 <d-1 f-2 a-3 c-5>
}

Fingersatzanweisungen können manuell oberhalb des Systems gesetzt werden, siehe Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.
Ausgewählte Schnipsel

Position von Fingersatz in Akkorden kontrollieren

Die Position von Fingersatzzahlen kann exakt kontrolliert werden.

```latex
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
}
```

Fingersatz auch innerhalb des Systems setzen

Normalerweise werden vertikal orientierte Fingersatzzahlen außerhalb des Systems gesetzt. Das kann aber verändert werden.

```latex
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 <g'0>}
```

Avoiding collisions with chord fingerings

Fingerings and string numbers applied to individual notes will automatically avoid beams and stems, but this is not true by default for fingerings and string numbers applied to the individual notes of chords. The following example shows how this default behavior can be overridden.

```latex
\relative c' {
  \set fingeringOrientations = #'(up)
  \set stringNumberOrientations = #'(up)
  \set strokeFingerOrientations = #'(up)

  % Default behavior
  r8
}
Kapitel 1: Musikalische Notation

<\!\<f c'\>8 >8
<\!\<f c'\>8 >8
<\!\<f c'\>8 >8

% No tweak needed
r8
<\!\<f c'\>8 >8
<\!\<f c'\>8 >8

% Corrected to avoid collisions
\override StrokeFinger.add-stem-support = ##t
<\!\<f c'\>8 >8 >8

\begin{music}
\relative { c'4 d \\
\hideNotes e4 f \\
\unHideNotes g a \\
\hideNotes b \\
\unHideNotes c \\
}
\end{music}

Siehe auch
Notationsreferenz: Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

Bekannte Probleme und Warnungen
Standardmäßig ist eine Zahl größer als 9 nicht unterstützt, wenn man die Schreibweise Note-Zahl einsetzt.

Unsichtbare Noten
Versteckte (oder unsichtbare oder transparente) Noten können sinnvoll sein, wenn man Notation für den Theorieunterricht oder Kompositionsübungen erstellen will.

\begin{music}
\relative { c''4 d \\
\hideNotes e4 f \\
\unHideNotes g a \\
\hideNotes b \\
\unHideNotes c \\
}
\end{music}
Notenköpfe, Hälse, Fähnchen und Pausen sind unsichtbar. Balken sind unsichtbar, wenn sie auf einer unsichtbaren Note beginnen. Objekte, die an unsichtbare Noten angehängt werden, sind trotzdem noch sichtbar.

\relative c' {  
e8(\p f g a)--  \\hideNotes  
e8(\p f g a)--  }

Vordefinierte Befehle
\hideNotes, \unHideNotes.

Siehe auch
Notationsreferenz: [Unsichtbare Pausen], Seite 54, Abschnitt 5.4.6 [Sichtbarkeit von Objekten], Seite 589, [Systeme verstecken], Seite 188.
Referenz der Interna: Abschnitt “Note_spacing_ engraver” in Referenz der Interna, Abschnitt “NoteSpacing” in Referenz der Interna.

Farbige Objekte
Einzelnen Objekten können einfach eigene Farben zugewiesen werden. Gültige Farbenbezeichnungen sind aufgelistet in Abschnitt A.7 [Liste der Farben], Seite 623.

\override NoteHead.color = #red  
c''4 c''  
\override NoteHead.color = #(x11-color 'LimeGreen)  
d''  
\override Stem.color = #blue  
e''

Die ganze Farbpalette, die für X11 definiert ist, kann mit der Scheme-Funktion x11-color benutzt werden. Diese Funktion hat ein Argument: entweder ein Symbol in der Form 'FooBar' oder eine Zeichenkette in der Form "FooBar". Die erste Form ist schneller zu schreiben und effizienter. Mit der zweiten Form ist es allerdings möglich, auch Farbbezeichnungen einzusetzen, die aus mehr als einem Wort bestehen.

Wenn x11-color die angegebene Farbbezeichnung nicht kennt, wird Schwarz eingesetzt.

\relative c' {  
\override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)  
\set Staff.instrumentName = \markup {  
\with-color #(x11-color 'navy) "Clarinet"}
Kapitel 1: Musikalische Notation

}\}
gis8 a
\override Beam.color = #(x11-color "medium turquoise")
gis a
\override Accidental.color = #(x11-color 'DarkRed)
gis a
\override NoteHead.color = #(x11-color "LimeGreen")
gis a
% this is deliberate nonsense; note that the stems remain black
\override Stem.color = #(x11-color 'Boggle)
b2 cis
}\}

\relative c'' {
\override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
   \with-color #(x11-color 'navy) "Clarinet"
}

\override Stem.color = #(rgb-color 0 0 0)
gis8 a
\override Stem.color = #(rgb-color 1 1 1)
gis8 a
\override Stem.color = #(rgb-color 0 0 0.5)
gis4 a
}\}

Siehe auch
Notationsreferenz: Abschnitt A.7 [Liste der Farben], Seite 623, Abschnitt 5.3.4 [Der tweak-Befehl], Seite 576.


Bekannte Probleme und Warnungen
Eine X11-Farbe hat nicht notwendigerweise exakt denselben Farbton wie eine ähnlich genannte normale Farbe.

Nicht alle X11-Farben lassen sich am Webbrowser erkennen, d. h. der Unterschied etwa zwischen LimeGreen und ForestGreen wird eventuell nicht dargestellt. Für die Benutzung im Internet wird die Benutzung von einfachen Farben nahegelegt (z. B. blue, green, red).

Noten in Akkorden können nicht mit \override eingefärbt werden, dazu muss \tweak benutzt werden. Siehe auch Abschnitt 5.3.4 [Der tweak-Befehl], Seite 576.
Klammern
Objekte können in Klammern gesetzt werden, indem vor ihnen der Befehl \parenthesize geschrieben wird. Wenn ein Akkord in Klammern gesetzt wird, wirkt sich das auf jede Note im Akkord aus. Innerhalb von einem Akkord gesetzte Befehle wirken sich auf einzelne Noten aus.

\relative {
  c''2 \parenthesize d
  c2 \parenthesize <c e g>
  c2 <c \parenthesize e g>
}

![Musiknotation mit Klammern]

Auch andere Objekte als Noten können in Klammern gesetzt werden. Wenn Artikulationszeichen in Klammern gesetzt werden sollen, braucht man ein Minuszeichen vor dem \parenthesize-Befehl.

\relative {
  c''2-\parenthesize -. d
  c2 \parenthesize r
}

![Musiknotation mit Klammern und Artikulationszeichen]

Siehe auch

Bekannte Probleme und Warnungen
Wenn man einen Akkord einklammert, wird um jede Note eine eigene Klammer gesetzt, anstatt den gesamten Akkord in eine große Klammer zu fassen.

Hälse
Immer, wenn das Programm eine Note findet, wird automatisch ein Notenhals (Abschnitt “Stem” in Referenz der Interna) -Objekt erzeugt. Auch für ganze Noten und Pausen werden sie erzeugt, aber unsichtbar gemacht.

Hälse können manuell gesetzt werden, um nach oben oder unten zu zeigen, siehe \undefined{Direction and placement}, Seite \undefined{undefined}.

Vordefinierte Befehle
\stemUp (Hälse nach oben), \stemDown (Hälse nach unten), \stemNeutral (Hälse je nach Notenposition).
Ausgewählte Schnipsel

Standardrichtung für Hälse auf der Mittellinie

Die Richtung von Hälsern auf der mittleren Linie kann mit der Stem-Eigenschaft neutral-direction gesetzt werden.

\relative c'' {
  a4 b c b
  \override Stem.neutral-direction = #up
  a4 b c b
  \override Stem.neutral-direction = #down
  a4 b c b
}

Siehe auch

Notationsreferenz: Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

1.7.2 Außerhalb des Notensystems

Dieser Abschnitt zeigt, wie man Elemente im System von außerhalb des Systems hervorhebt.

Erklärungen in Ballonform

Notationselemente können bezeichnet und markiert werden, indem um sie eine rechteckige Blase gezeichnet wird. Dies ist vor allem dazu da, Notation zu erklären.

\new Voice \with { \consists "Balloon_engraver" }
\relative c'' {
  \balloonGrobText #'Stem #'(3 . 4) \markup { "I'm a Stem" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "I'm a rest" }
  r
  <c, g'\balloonText #'(-2 . -2) \markup { "I'm a note head" } c>2.
}

Es gibt zwei Funktionen, balloonGrobText und balloonText; die erste wird auf gleiche Art wie ein \once \override eingesetzt und Text an einen Grob zu hängen, die zweite funktioniert wie ein \tweak und wird üblicherweise innerhalb von Akkorden eingesetzt, um Text an einzelne Noten zu hängen.
Textblasen beeinflussen normalerweise die Positionierung der Notation, aber das kann geändert werden.

\new Voice \with { \consists "Balloon_engraver" }
{
  \balloonLengthOff
  \balloonGrobText #'Stem #'(3 . 4) \markup { "I'm a Stem" }
  a8
  \balloonGrobText #'Rest #'(−4 . −4) \markup { "I'm a rest" }
  r
  \balloonLengthOn
  <c, g'-\balloonText #'(−2 . −2) \markup { "I'm a note head" } c>2.
}

\layout {
  \context {
    \Staff
    \consists "Grid_point_engraver"
    \gridInterval = #(ly:make-moment 1/4)
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
  }
}

\score {
  \new ChoirStaff <<

Vordefinierte Befehle
\balloonLengthOn, \balloonLengthOff.

Siehe auch


Gitternetzlinien
Vertikale Linien können zwischen Systemen gesetzt werden, die mit den Noten synchronisiert sind.


\layout {
  \context {
    \Staff
    \consists "Grid_point_engraver"
    \gridInterval = #(ly:make-moment 1/4)
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
  }
}

\score {
  \new ChoirStaff <<
Ausgewählte Schnipsel

Gitternetzlinien: Aussehen verändern

Die Erscheinung der Gitternetzlinien kann durch einige Eigenschaften geändert werden.

\score {
   \new ChoirStaff <<
      \new Staff {
         \relative c'' {
            \stemUp
            c''4. d8 e8 f g4
         }
      }
      \new Staff {
         \relative c {
            % this moves them up one staff space from the default position
            \override Score.GridLine.extra-offset = #'(0.0 . 1.0)
            \stemDown
            \clef bass
            \once \override Score.GridLine.thickness = #5.0
            c4
            \once \override Score.GridLine.thickness = #1.0
            g'4
            \once \override Score.GridLine.thickness = #3.0
            f4
            \once \override Score.GridLine.thickness = #5.0
            e4
         }
      }
   >>
}

\layout {
   \context {
      \Staff
% set up grids
\consists "Grid_point_engraver"
% set the grid interval to one quarter note
gridInterval = #(ly:make-moment 1/4)
}
\context {\Score
\consists "Grid_line_span_engraver"
% this moves them to the right half a staff space
\override NoteColumn.X-offset = #-0.5
}
}
}

Siehe auch


Analyseklammern


\layout {
\context {\Voice
\consists "Horizontal_bracket_engraver"
}
}\relative {
c'2\startGroup
d\stopGroup
}

Analyseklammern können verschachtelt sein.

\layout {
Siehe auch


1.8 Text

### 1.8.1 Text eingeben

Dieser Abschnitt zeigt verschiedene Arten, wie Text in die Partitur eingefügt werden kann.

**Achtung:** Wenn man Zeichen mit Akzenten und Umlaute oder besondere Zeichen (wie etwa Text mit anderen Alphabeten) einfügen möchte, kann man die Zeichen einfach direkt in die Datei einfügen. Die Datei muss als UTF-8 gespeichert werden. Für mehr Information siehe [Zeichenkodierung], Seite 482.

**Textarten**

Am einfachsten kann Text mit geraden Anführungsstrichen in eine Partitur eingefügt werden, wie das folgende Beispiel zeigt. Derartiger Text kann manuell über oder unter dem Notensystem platziert werden, die Syntax hierzu ist beschrieben in Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581.

\relative { a'8^"pizz." g f e a4-"scherz." f }

Diese Syntax ist eine Kurzform, komplexere Formatierungen können einem Text hinzugefügt werden, wenn man explizit den \markup-Befehl mit darauf folgenden geschweiften Klammern einsetzt, wie beschrieben in Abschnitt 1.8.2 [Text formatieren], Seite 223.

\relative { a'8\markup { \italic pizz. } g f e a4\markup { \tiny scherz. \bold molto } f }

Standardmäßig haben Textbeschriftungen keinen Einfluss auf die Positionierung der Noten. Man kann aber auch bestimmen, dass die Breite des Textes mit berücksichtigt wird. Im nächsten Beispiel fordert der erste Text keinen Platz, während der zweite die Note nach rechts verschiebt. Das Verhalten wird mit dem Befehl \textLengthOn (Textlänge an) erreicht, rückgängig kann es mit dem Befehl \textLengthOff gemacht werden.

\relative { a'8^"pizz." g f e }
Kapitel 1: Musikalische Notation


Vordefinierte Befehle
\textLengthOn, \textLengthOff.

Siehe auch
Notationsreferenz: Abschnitt 1.8.2 [Text formatieren], Seite 223, Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581, [Artikulationszeichen und Verzierungen], Seite 113.

Bekannte Probleme und Warnungen
Eine Überprüfung, ob sich auch alle Textbeschriftungen und Gesangstext innerhalb der Ränder der Noten befinden, braucht verhältnismäßig viel Rechenaufwand. Sollten Sie aber etwas bessere Leistung bevorzugen, schreiben Sie in Ihre Datei:
\override Score.PaperColumn.keep-inside-line = ##f

Text mit Verbindungslinien
Einige Aufführungsanweisungen, etwa rallentando oder accelerando, werden als Text geschrieben, gefolgt von einer gestrichelten Linie, die anzeigt, wie weit sich die Anweisung auswirkt. Solche Objekte, „Strecker“ (engl. spanners) genannt, können von einer Note bis zu einer anderen mit folgender Anweisung erstellt werden:
\relative{
  \override TextSpanner.bound-details.left.text = "rit."
  b'1\startTextSpan
  e,\stopTextSpan
}

Der Text wird durch Objekteigenschaften beeinflusst. In den Standardeinstellungen wird er kursiv ausgegeben, aber eine andere Formatierung kann erreicht werden, indem man \markup-Blöcke einsetzt, wie beschrieben in Abschnitt 1.8.2 [Text formatieren], Seite 223.
\relative {
  \override TextSpanner.bound-details.left.text = "rit."
}
Kapitel 1: Musikalische Notation


Vordefinierte Befehle

\textSpannerUp, \textSpannerDown, \textSpannerNeutral.

Ausgewählte Schnipsel

_Dynamiktextstrecker nachgestellt_

Die cresc, dim und decresc Strecker können umdefiniert werden, um nachgestellt zu funktionieren und einen Textstrekker zu produzieren. Eigene Strecker können auch einfach definiert werden. Klammer- und Tex crescendi können einfach vermischt werden. \( < \) und \( > \) erstellen normalerweise Klammern, \( \text{cresc} \) usw. dagegen normalerweise Textspanner.

\texttt{% Some sample text dynamic spanners, to be used as postfix operators}
\texttt{crpoco =}
\texttt{
#(make-music 'CrescendoEvent
 'span-direction START
 'span-type 'text
 'span-text "cresc. poco a poco")}

\texttt{\relative c' {}
 c4\cresc d4 e4 f4 |
 g4 a4! b4\crpoco c4 |
 c4 d4 e4 f4 |
 g4 a4! b4\< c4 |
 g4\dim a4 b4\decresc c4!}

_Eigene Dynamiktextspanner nachgestellt_

Die Nachstellung funktioniert für eigene Crescendo-Textstrecken. Die Strecker sollten an der ersten Note eines Taktes beginnen. Man muss \texttt{-mycresc} benutzen, sonst wird der Beginn des Streckers der nächsten Note zugewiesen.

\texttt{% Two functions for (de)crescendo spanners where you can explicitly give the % spanner text.}
\texttt{mycresc =}
\texttt{
#(define-music-function (mymarkup) (markup?)
 (make-music 'CrescendoEvent
 'span-direction START
 'span-type 'text
 'span-text "cresc. poco a poco")
 (make-music 'DecrescendoEvent
 'span-direction END
 'span-type 'text
 'span-text "decresc."
 'span-text "dim."
 'span-text "decresc")
}
Textartige Zeichen

Verschiedene Textelemente können der Partitur hinzugefügt werden, indem man die Syntax für Zeichen einsetzen, wie beschrieben in [Übungszeichen], Seite 102:

\relative {
  c''4
  \mark "Allegro"
  c c c
}

Diese Syntax ermöglicht es, beliebigen Text über eine Taktlinie zu platzieren, weitere Formatierungs möglichkeiten sind mit dem \markup-Befehl gegeben, wie beschrieben in Abschnitt 1.8.2 [Text formatieren], Seite 223:

\relative {
  <c' e>1
  \mark \markup { \italic { colla parte } }
  <d f>2 <e g>
  <c f aes>1
}
Diese Syntax ermöglicht es auch, besondere Zeichen einzufügen, wie etwa Coda-, Segno- oder Fermatenzeichen, indem das entsprechende Symbol mit dem Befehl `\musicglyph` angegeben wird, wie beschrieben in [Musikalische Notation innerhalb einer Textbeschriftung], Seite 234:

```
\relative { 
  <bes' f>2 <aes d> 
  \mark \markup { \musicglyph "scripts.ufermata" } 
  <e g>1 
}
```


```
\mark "Allegro"
\relative c'' { 
  g2 c 
  \once \override Score.RehearsalMark.break-visibility = #end-of-line-visible 
  \once \override Score.RehearsalMark.self-alignment-X = #RIGHT 
  \mark "D.C. al Fine" 
  \break 
  g2 b, 
  c1 \bar "||"
}
```

Ausgewählte Schnipsel

*Printing marks at the end of a line*

Marks can be printed at the end of the current line, instead of the beginning of the following line. In such cases, it might be preferable to align the right end of the mark with the bar line.
Kapitel 1: Musikalische Notation

Zeichen über jedem System ausgeben

Normalerweise werden Textzeichen nur über dem obersten Notensystem gesetzt. Sie können aber auch über jedem System ausgegeben werden.

\score {
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
    \context {
      \Staff
      \consists "Mark_engraver"
      \consists "Staff_collecting_engraver"
    }
  }
}

Siehe auch

Notationsreferenz: [Übungszeichen], Seite 102, Abschnitt 1.8.2 [Text formatieren], Seite 223, [Musikalische Notation innerhalb einer Textbeschriftung], Seite 234, Abschnitt A.8 [Die Emmentaler-Schriftart], Seite 625.


Bekannte Probleme und Warnungen

Wenn ein Zeichen am Ende des letzten Taktes einer Partitur gesetzt wird (wenn also keine nächste Zeile mehr kommt), wird das Zeichen nicht ausgegeben.
Kapitel 1: Musikalische Notation

Separater Text
Eine \markup-Umgebung kann auch für sich alleine existieren, außerhalb einer \score-Umgebung, als ein Ausdruck auf der höchsten Ebene. Diese Syntax ist beschrieben in Abschnitt 3.1.5 [Die Dateistruktur], Seite 451.

\markup{
Morgen, morgen, und morgen...
}

Morgen, morgen, und morgen...


\score{
\timesig.C44\clefs.G\noteheads.s0
\begin{columns}
\begin{column}
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
\end{column}
\begin{column}
\line{Morgen, übermorgen, und überübermorgen...}
\end{column}
\end{columns}
}\score\markup{
Morgen, übermorgen, und überübermorgen...
}

Unabhängige Textabschnitte können über mehrere Seiten reichen, so dass man Textdokumente oder Bücher ausschließlich mit LilyPond setzen kann. Einzelheiten zu den vielfältigen Möglichkeiten finden sich in [Textbeschriftung über mehrere Seiten], Seite 236.

Vordefinierte Befehle
\markup, \markuplist.

Ausgewählte Schnipsel
Isolierter Text in zwei Spalten
Isolierter Text kann in mehreren Spalten mit \markup-Befehlen angeordnet werden:
\markup{
\fill-line{
\hspage #1
\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\column{
\line{Morgen, übermorgen, und überübermorgen...}
}\column{
\line{O sacrum convivium}
\line{in quo Christus sumitur,}
\line{recolitur memoria passionis ejus,}
}\end{columns}
}\markup
Kapitel 1: Musikalische Notation

O sacrum convivium in quo Christus sumitur, recolitur memoria passionis ejus, mens impletur gratia, futurae gloriae nobis pignus datur. Amen.

O sacred feast in which Christ is received, the memory of His Passion is renewed, the mind is filled with grace, and a pledge of future glory is given to us. Amen.

Siehe auch

Notationsreferenz: Abschnitt 1.8.2 [Text formatieren], Seite 223, Abschnitt 3.1.5 [Die Dateistruktur], Seite 451, Abschnitt 3.1.2 [Mehrere Partituren in einem Buch], Seite 448, [Textbeschriftung über mehrere Seiten], Seite 236.


1.8.2 Text formatieren

Dieser Abschnitt zeigt grundlegende und fortgeschrittene Formatierung von Text, wobei der Textbeschriftungsmodus (\markup) benutzt wird.

Textbeschriftung (Eindeutung)

Eine \markup-Umgebung wird benutzt, um Text mit einer großen Anzahl von Formatierungs möglichkeiten (im „markup-Modus“) zu setzen.

Die Syntax für Textbeschriftungen ähnelt der normalen Syntax von LilyPond: ein \markup-Ausdruck wird in geschweifte Klammern eingeschlossen ( {... }). Ein einzelnes Wort wird als ein Minimalausdruck erachtet und muss deshalb nicht notwendigerweise eingeklammert werden.

Anders als Text in Anführungsstrichen können sich in einer Textbeschriftungs umgebung (\markup) geschachtelte Ausdrücke oder weitere Textbefehle befinden, eingeführt mit einem Backslash (\). Derartige Befehle beziehen sich nur auf den ersten der folgenden Ausdrücke.

\relative {
a'1-'\markup intenso
a2-'\markup { poco \italic più forte }
c e1
d2-'\markup { \italic "string. assai" }
Eine \texttt{\textbackslash markup}-Umggebung kann auch Text in Anführungszeichen beinhalten. Derartige Zeichenketten werden als ein Textausdruck angesehen, und darum werden innerhalb von ihnen Befehle oder Sonderzeichen (wie \textbackslash oder \#) so ausgegeben, wie sie eingeben werden. Doppelte Anführungsstriche können gesetzt werden, indem man ihnen einen Backslash voranstellt.

\begin{verbatim}
\relative {
    a'1^"italic Text..."
    a \markup { \italic "... setzt \"kursive\" Buchstaben!" }
    a a
}
\end{verbatim}

Damit eine Anzahl von Wörtern als ein einziger Ausdruck behandelt wird, müssen alle Wörter zwischen geraden Anführungszeichen (Shift+2) stehen oder ihnen muss ein Befehl vorangestellt werden. Die Art, wie die Ausdrücke definiert sind, wirkt sich darauf aus, wie sie übereinander gestapelt, mittig und aneinander ausgerichtet werden. Im folgenden Beispiel verhält sich der zweite \texttt{\textbackslash markup}-Ausdruck genauso wie der erste:

\begin{verbatim}
\relative c'' {
    \makeBoxes{\relative a'1^"center-column { a bbb c }"
                \relative c1^"center-column { a { bbb c } }
                \relative c1^"center-column { a \line { bbb c } }
                \relative c1^"center-column { a "bbb c" }
    }
    a a
    bbb bbb a a
    c c bbb c bbb c
}
\end{verbatim}

Textbeschriftung kann auch durch Variablen definiert werden. Diese Variablen können dann direkt an Noten angefügt werden:

\begin{verbatim}
allegro = \markup { \texttt{\textbackslash bold} \texttt{\textbackslash large} Allegro }
\end{verbatim}
Eine ausführliche Liste der \markup-Befehle findet sich in Abschnitt A.10 [Textbeschriftungsbefehle], Seite 646.

Siehe auch

Notationsreferenz: Abschnitt A.10 [Textbeschriftungsbefehle], Seite 646.
Installierte Dateien: scm/markup.scm

Bekannte Probleme und Warnungen

Syntaxfehler im Textbeschriftungsmodus können sehr verwirrend sein.

Überblick über die wichtigsten Textbeschriftungsbefehle

Einfache Änderungen des Schriftartschnitts können im Textbeschriftungsmodus vorgenommen werden:

\relative {  
  d'\markup {  
    \bold { Più mosso }  
    \italic { non troppo \underline Vivo }  
  }  
  r2 r4 r8  
  d,\markup { \italic quasi \smallCaps Tromba }  
  f1 d2 r  
}

**Più mosso non troppo Vivo**

Die Schriftgröße kann auf verschiedene Arten verändert werden, relativ zur globalen Notensystemgröße:

Sie kann auf eine vordefinierte Größe gesetzt werden:

\relative b' {  
  b1\markup { \huge Sinfonia }  
  b1\markup { \teeny da }  
  b1-\markup { \normalsize camera }  
}

**Sinfonia**

Sie kann relativ zum vorherigen Wert gesetzt werden:

\relative b' {  
  b1\markup { \larger Sinfonia }  
  b1\markup { \smaller da }  
  b1-\markup { \magnify #0.6 camera }  
}
Sie kann vergrößert oder verkleinert werden relativ zum Wert, der von der globalen Noten-systemgröße vorgegeben wird:

\relative b' {  
  b1_\markup { \fontsize #-2 Sinfonia }  
  b1^\markup { \fontsize #1 da }  
  b1-\markup { \fontsize #3 camera }  
}\n
Sie kann auch auf eine bestimmte Punktegröße festgelegt werden, unabhängig von der globalen Notensystemgröße:

\relative b' {  
  b1_\markup { \abs-fontsize #20 Sinfonia }  
  b1^\markup { \abs-fontsize #8 da }  
  b1-\markup { \abs-fontsize #14 camera }  
}\n
Text kann auch hoch- bzw. tiefgestellt gesetzt werden. Die so markierten Buchstaben werden automatisch in einer kleineren Schriftgröße gesetzt, aber die normale Schriftgröße kann auch eingesetzt werden:

\markup {  
  \column {  
    \line { 1 \super st movement }  
    \line { 1 \normal-size-super st movement  
      \sub { (part two) } }  
  }  
}\n
1 st movement  
1 st movement (part two)


\markup {  
  \column {  
    \line { Act \number 1 }  
    \line { \sans { Scene I. } }  
    \line { \typewriter { Verona. An open place. } }  
  }  
}
\line { Enter \roman Valentine and Proteus. }
}
}

Act 1
Scene I.
Verona. An open place.
Enter Valentine and Proteus.

Einige dieser Schriftarten, etwa die Zahlenschriftart oder die Schriftart für Dynamikzeichen, stellen nicht alle Zeichen zur Verfügung, wie beschrieben in [Neue Lautstärkezeichen], Seite 121, und [Manuelle Wiederholungszeichen], Seite 143.

Einige Schriftartbefehle können ungewollte Leerzeichen innerhalb von Wörtern hervorrufen. Das kann vermieden werden, indem die einzelnen Elemente mit dem Befehl \concat zu einem Element verschmolzen werden:
\markup {
\column {
\line {
\concat { 1 \super st } 
\move
}
\line {
\concat { \dynamic p , } 
\italic { con dolce espressione } 
}
}
}

1st movement
\p, con dolce espressione

Eine ausführliche Liste der unterschiedlichen Befehl zur Beeinflussung der Schriftarten findet sich in Abschnitt A.10.1 [Font], Seite 646.
Es ist auch möglich, eigene Schriftfamilien zu definieren, wie erklärt in Abschnitt 1.8.3 [Schriftarten], Seite 237.

Vordefinierte Befehle
\teeny, \tiny, \small, \normalsize, \large, \huge, \smaller, \larger.

Siehe auch
Notationsreferenz: Abschnitt A.10.1 [Font], Seite 646, [Neue Lautstärkezeichen], Seite 121, [Manuelle Wiederholungszeichen], Seite 143, Abschnitt 1.8.3 [Schriftarten], Seite 237.
Installierte Dateien: scm/define-markup-commands.scm.

Bekannte Probleme und Warnungen
Wenn die Befehle \teeny, \tiny, \small, \normalsize, \large und \huge eingesetzt werden, erhält man schlechte Zeilenabstände verglichen mit \fontsize.
Textausrichtung


Textbeschriftungsobjekte können auf verschiedene Weise ausgerichtet werden. Standardmäßig wird ein Textobjekt an seiner linken Ecke ausgerichtet, darum wird das erste und zweite Objekt gleichermaßen an der linken Ecke ausgerichtet.

\relative {
  d''1-\markup { poco }
  f
  d-\markup { \left-align poco }
  f
  d-\markup { \center-align { poco } }
  f
  d-\markup { \right-align poco }
}

Die horizontale Ausrichtung kann mit einer Zahl auf einen exakten Wert festgelegt werden:

\relative {
  a'1-\markup { \halign #-1 poco }
  e'
  a,-\markup { \halign #0 poco }
  e'
  a,-\markup { \halign #0.5 poco }
  e'
  a,-\markup { \halign #2 poco }
}

Manche Objekte haben eigene Ausrichtungsvorgänge und werden deshalb nicht von diesen Befehlen beeinflusst. Es ist möglich, solche Objekte als eine Einheit anzusprechen und zu bewegen, wie gezeigt in [Textartige Zeichen], Seite 219.

Die vertikale Ausrichtung ist etwas schwieriger. Textelemente können komplett verschoben werden, es ist aber auch möglich, nur einen Teil innerhalb der Textbeschriftung zu bewegen. In diesem Fall muss dem zu verschiebenden Objekt ein Ankerpunkt zugewiesen werden, welcher entweder ein anderes Textelement oder ein unsichtbares Objekt sein kann (im Beispiel mit \null erstellt). Der letzte Text im Beispiel hat keinen Anker und wird deshalb auch nicht verschoben.

\relative {
  d'2-\markup {
    Acte I
    \raise #2 { Scène 1 }
  }
  a'
}
Einige Befehle können sowohl die horizontale als auch die vertikale Ausrichtung von Textobjekten beeinflussen. Jedes Objekt, das auf diese Weise verschoben wird, benötigt einen Anker:

```latex
\relative {
  d'2\markup {
    \ActeI
    \translate #'(-1 . 2) "Scène 1"
  }
  a'
  g\markup {
    \null
    \general-align #Y #3.2 \bold "Très modéré"
  }
  a
  d,\markup {
    \null
    \translate-scaled #'(-1 . 2) \teeny "Une forêt."
  }
  a'4 a g2 a
}
```

Ein Textbeschriftungsobjekt kann mehrere Zeilen beinhalten. Im folgenden Beispiel wird jeder Ausdruck innerhalb von `\markup` auf einer eigenen Zeile gesetzt, entweder linksbündig oder zentriert:

```latex
\markup {
  \column {
    a
    "b c"
    \line { d e f }
  }
}```
Eine Anzahl an Ausdrücken innerhalb von \textup{markup} kann auch gestreckt werden, so dass die gesamte Seitenbreite benutzt wird. Wenn nur ein Objekt vorhanden ist, wird es zentriert gesetzt. Die Ausdrücke selber können wiederum mehrzeilig sein und andere Textbeschriftungsbefehle beinhalten.

\textup{ markup } \{  
  \text{ fill-line } \{ 
    \text{ line } \{ \text{ William S. Gilbert } \}  
    \text{ center-column } \{ 
      \text{ huge } \textup{ smallCaps } \text{ "The Mikado" } 
      \text{ or } 
      \text{ smallCaps } \text{ "The Town of Titipu" } 
    \}  
    \text{ line } \{ \text{ Sir Arthur Sullivan } \}  
  \}  
\text{ markup } \{  
  \text{ fill-line } \{ \text{ 1885 } \}  
\}  

\text{ William S. Gilbert } \quad \text{ THE MIKADO } \quad \text{ Sir Arthur Sullivan }  
\text{ or } \quad \text{ THE TOWN OF TITIPU }  
\text{ 1885 }  

Längere Texte können auch automatisch umgebrochen werden, wobei es möglich ist, die Zeilenbreite zu bestimmen. Der Text ist entweder linksbündig oder im Blockssatz, wie das nächste Beispiel illustriert:

\textup{ markup } \{  
  \text{ column } \{ 
    \text{ line } \textup{ smallCaps } \text{ La vida breve } \}  
    \text{ line } \textup{ bold } \text{ Acto I } \}  
\text{ wrap } \textup{ italic } \{ 
  (\text{ La escena representa el corral de una casa de gitanos en el Albaicín de Granada. Al fondo una puerta por la que se ve el negro interior de una Fragua, iluminado por los rojos resplandores del fuego.})
Kapitel 1: Musikalische Notation

La vida breve

Acto I

(La escena representa el corral de una casa de gitanos en el Albaicín de Granada. Al fondo una puerta por la que se ve el negro interior de una Fragua, iluminado por los rojos resplandores del fuego.)

Acto II

(Calle de Granada. Fachada de la casa de Carmela y su hermano Manuel con grandes ventanas abiertas a través de las que se ve el patio donde se celebra una alegre fiesta)

Eine vollständige Liste der Textausrichtungsbefehle findet sich in Abschnitt A.10.2 [Align], Seite 656.

Siehe auch

Notationsreferenz: Abschnitt A.10.2 [Align], Seite 656, [Textartige Zeichen], Seite 219.
Installierte Dateien: scm/define-markup-commands.scm.

Graphische Notation innerhalb einer Textbeschriftung

Verschiedene graphische Objekte können im Textbeschriftungsmodus eingefügt werden.
Mit bestimmten Textbeschriftungsbefehlen kann man Textelementen Graphik hinzufügen, wie das nächste Beispiel zeigt:

\markup \fill-line {
  \center-column {
    \circle Jack
    \box "in the box"
    \null
    \line {
      Erik Satie
      \hspace #3
      \bracket "1866 - 1925"
    }
  }
}
Es kann nötig sein, einem Text mehr Platz einzuräumen. Das geschieht mit verschiedenen Befehlen, wie das folgende Beispiel zeigt. Eine ausführliche Übersicht findet sich in Abschnitt A.10.2 [Align], Seite 656.

\markup \fill-line {
  \center-column {
    \box "Charles Ives (1874 - 1954)"
    \null
    \box \pad-markup #2 "THE UNANSWERED QUESTION"
    \box \pad-x #8 "A Cosmic Landscape"
    \null
  }
}
\markup \column {
  \line {
    \hspace #10
    \box \pad-to-box #'(-5 . 20) #'(0 . 5)
    \bold "Largo to Presto"
  }
  \pad-around #3
    "String quartet keeps very even time,
     Flute quartet keeps very uneven time."
}

<table>
<thead>
<tr>
<th>Charles Ives (1874 - 1954)</th>
</tr>
</thead>
<tbody>
<tr>
<td>THE UNANSWERED QUESTION</td>
</tr>
<tr>
<td>A Cosmic Landscape</td>
</tr>
</tbody>
</table>

Largo to Presto

String quartet keeps very even time, Flute quartet keeps very uneven time.
Andere graphische Elemente oder Symbole können gesetzt werden, ohne dass man Text benötigt. Wie mit allen Textbeschriften können Objekte innerhalb von \markup kombiniert werden.

\markup {
  \combine
  \draw-circle #4 #0.4 ##f
  \filled-box #'(-4 . 4) #'(-0.5 . 0.5) #1
  \hspace #5

  \center-column {
    \triangle ##t
    \combine
      \draw-line #'(0 . 4)
      \arrow-head #Y #DOWN ##f
  }
}

Fortgeschrittene graphische Möglichkeiten bietet unter Anderem eine Funktion, mit der man externe Graphiken im Encapsulated PostScript (eps) -Format einbinden kann oder aber Graphiken direkt in den Quelltext unter Verwendung von PostScript-Code notiert. In diesem Fall kann es nötig sein, die Größe der Zeichnung explizit anzugeben, wie im Beispiel unten gezeigt:

c'1^\markup {
  \combine
    \epsfile #X #10 "/context-example.eps"
    \with-dimensions #'(0 . 6) #'(0 . 10)
    \postscript "
    -2 3 translate
    2.7 2 scale
    newpath
    2 -1 moveto
    4 -2 4 1 1 arct
    4 2 3 3 1 arct
    0 4 0 3 1 arct
    0 0 1 -1 1 arct
    closepath
    stroke"
}

c'

Eine ausführliche Liste der Graphik-Befehle findet sich in Abschnitt A.10.3 [Graphic], Seite 672.

Musikalische Notation innerhalb einer Textbeschriftung

Auch Musikobjekte können innerhalb der Textbeschriftungsumgebung gesetzt werden.

Noten und Versetzungszeichen lassen sich mit \markup eingefügt:

\begin{verbatim}
a'2 a'\markup {
   \note "4" #1
= \note-by-number #1 #1 #1.5}
b'1_\markup {
   \natural \semiflat \flat \sesquiflat \doubleflat}
\glissando
a'_1_\markup {
   \natural \semisharp \sharp \sesquisharp \doublesharp}
\glissando b'
\end{verbatim}

Andere Notationsobjekte können auch eingefügt werden:

\begin{verbatim}
g1 bes
ees-\markup {
   \finger 4 \tied-lyric "-"
   \finger 1}
fis_\markup { \dynamic rf }
bes^-\markup {
   \beam #8 #0.1 #0.5}
cis
d-\markup {
   \markalphabet #8 \markletter #8}
\end{verbatim}
Allgemeiner gesagt kann jedes verfügbare Notationssymbol unabhängig von der Notation als ein Textbeschriftungsobjekt eingefügt werden, wie unten gezeigt. Eine vollständige Liste der verfügbaren Symbole findet sich in Abschnitt A.8 [Die Emmentaler-Schriftart], Seite 625.

\relative {
  c''2
  \markup { \musicglyph "eight" }
  c,4
  c,8.. \markup { \musicglyph "clefs.G_change" }
  c16
  c2" \markup { \musicglyph "timesig.neomensural94" }
}

Eine andere Möglichkeit, andere als Textsymbole zu schreiben, findet sich in [Was sind Schriftarten], Seite 237. Diese Methode bietet sich an, um Klammern unterschiedlicher Größe zu setzen.

Der Textbeschriftungsmodus unterstützt auch Diagramme für bestimmte Instrumente:

\relative {
  c'\'1\markup {
    \fret-diagram-terse "x;x;o;2;3;2;"
  }
  c^-\markup {
    \harp-pedal "\--v|--ov--"
  }
  c
  c^-\markup {
    \combine
    \musicglyph "accordion.discant"
    \combine
    \raise #0.5 \musicglyph "accordion.dot"
    \raise #1.5 \musicglyph "accordion.dot"
  }
}

Derartige Diagramme sind dokumentiert in Abschnitt A.10.5 [Instrument Specific Markup], Seite 686.
Sogar eine ganze Partitur kann in ein Textbeschriftungsobjekt eingefügt werden. In diesem Fall muss die eingefügte \score-Umgebung eine \layout-Umgebung haben, wie in diesem Beispiel:

\relative {
  c'4 d\markup {
    \score {
      \relative { c'4 d e f }
      \layout { }
    }
  }
  e f |
  c d e f
}

\begin{musicfigure}
\includegraphics[scale=0.5]{musicexample.png}
\end{musicfigure}

Eine vollständige Liste der Musiksymbol-Befehle findet sich in Abschnitt A.10.4 [Music], Seite 680.

Siehe auch

Notationsreferenz: Abschnitt A.10.4 [Music], Seite 680, Abschnitt A.8 [Die Emmentaler-Schriftart], Seite 625, [Was sind Schriftarten], Seite 237.


Textbeschriftung über mehrere Seiten

Normale Textbeschriftungsobjekte können nicht getrennt werden, aber mit einer spezifischen Umgebung ist es möglich, Text auch über mehrere Seiten fließen zu lassen:

\markuptlist {
  \justified-lines {
    A very long text of justified lines.
    ...
  }
  \wordwrap-lines {
    Another very long paragraph.
    ...
  }
  ...
}

A very long text of justified lines. ...

Another very long paragraph. ...
... 

Die Syntax braucht eine Liste von Textbeschriftungen folgender Art:
• das Resultat eines Beschriftungsbefehls,
• eine Textbeschriftungsliste,
• eine Liste von Beschriftungsbefehlen.

Eine vollständige Liste der Beschriftungsbefehle findet sich in Abschnitt A.11 \[Textbeschriftungsbefehle\], Seite 701.

Siehe auch
Notationsreferenz: Abschnitt A.11 \[Textbeschriftungsbefehle\], Seite 701.
Erweitern: Abschnitt “Neue Definitionen von Beschriftungsbefehlen” in \textit{Extending}.
Installierte Dateien: \texttt{scm/define-markup-commands.scm}.
Schnipsel: Abschnitt “Text” in \texttt{Schnipsel}.
Referenz der Interna: Abschnitt “TextScript” in \textit{Referenz der Interna}.

Vordefinierte Befehle
\texttt{\markuplist}.

1.8.3 Schriftarten
Dieser Abschnitt zeigt, wie Schriftarten eingesetzt werden können und wie man sie in Partituren ändern kann.

Was sind Schriftarten
Schriftarten werden von mehreren Bibliotheken verwaltet. FontConfig wird benutzt, um die vorhandenen Schriftarten des Systems zu erkennen, die gewählte Schriftart wird dann mit Pango verarbeitet.

Notationsschriftarten können als eine Ansammlung von besonderen Zeichen erklärt werden, wobei die Sonderzeichen in verschiedene Familien klassifiziert werden. Die Syntax des folgenden Beispiels ermöglicht es, direkt auf verschiedene nicht textuelle Sonderzeichen der \texttt{feta-Glyphe} zuzugreifen. Das ist die Standardschriftart für Notationselemente in LilyPond.

\begin{verbatim}
\markup { 
  \vcenter { 
    \override #'(font-encoding . fetaBraces) 
    \lookup "brace120" 
    \override #'(font-encoding . fetaText) 
    \column { 1 3 sf } 
    \override #'(font-encoding . fetaMusic) 
    \lookup "noteheads.s0petrucci"
  }
}
\end{verbatim}

\begin{figure}[h]
\centering
\includegraphics[width=0.2\textwidth]{example.png}
\caption{Example notation with special characters using \texttt{feta-Glyphe}.}
\end{figure}
Außer den verschiedenen Klammern, die in \texttt{fetaBraces} in verschiedenen Größen enthalten sind, lassen sich alle dieses Symbole auch mit einer einfacheren Syntax notieren. Sie ist beschrieben in [Musikalische Notation innerhalb einer Textbeschriftung], Seite 234.

Wenn man die Klammern von \texttt{fetaBraces} benutzt, wird die Größe der Klammer durch einen numeralen Part in der Bezeichnung des Glyphs bestimmt. Als Wert kann eine Ganzzahl von 0 bis 575 benutzt werden, wobei 0 die kleinste Klammern ergibt. Der optimale Wert muss durch Ausprobieren herausgefunden werden. Diese Glyphen sind alle linke Klammern, rechte Klammern lassen sich durch eine Drehung herstellen, siehe Abschnitt 5.4.8 [Drehen von Objekten], Seite 595.

Drei Textschriftarten sind verfügbar (auf Englisch \texttt{family} genannt): mit \texttt{roman} eine Schriftart mit Serifen (Standard ist New Century Schoolbook), mit \texttt{sans} eine serifenlose (gerade) Schriftart und mit \texttt{typewriter} eine Schreibmaschinenchrift, in welcher die Buchstaben alle die gleiche Weite haben. Die aktuelle Schriftart von \texttt{sans} und \texttt{typewriter} wird durch Pango entsprechend den Systemvorgaben gewählt.

Jede Familie kann verschiedene Schriftschnitte besitzen. Im Englischen wird unterschieden zwischen \texttt{shape} für kursive Schnitte und \texttt{series} für fette Schnitte. Im folgenden Beispiel wird demonstriert, wie man die verschiedenen Eigenschaften auswählen kann. Der Wert, der \texttt{font-size} übergeben wird, entspricht der geforderten Änderung in Bezug auf die Standardschriftgröße.

\begin{verbatim}
\override Score.RehearsalMark.font-family = #'typewriter
\markup "Ouverture"
\override Voice.TextScript.font-shape = #'italic
\override Voice.TextScript.font-series = #'bold
c"4"smaller
\markup {\em \override #'(font-shape . italic)
\override #'(font-size . 4)
Idomeneo,}
\markup {\override #'(font-family . typewriter)
\override #'(font-series . bold)
re
di
}\markup #'(font-family . sans)
\override #'(font-size . 4)
Creta
\end{verbatim}

Eine ähnliche Syntax kann im Textbeschriftungsmodus eingesetzt werden, hier bietet es sich aber an, die einfacheren Befehle zu verwenden, die erklärt wurden in [Überblick über die wichtigsten Textbeschriftungsbefehle], Seite 225:
Idomeneo,  
re di Creta

Auch wenn es einfach ist, zwischen den vordefinierten Schriftarten umzuschalten, kann man auch eigene Schriftarten verwenden, wie erklärt in folgenden Abschnitten: [Schriftarten für einen Eintrag], Seite 239, und [Schriftart des gesamten Dokuments], Seite 239.

Siehe auch
Notationsreferenz: Abschnitt A.8 [Die Emmentaler-Schriftart], Seite 625, Abschnitt 5.4.8 [Drehen von Objekten], Seite 595, [Musikalische Notation innerhalb einer Textbeschriftung], Seite 234, [Überblick über die wichtigsten Textbeschriftungsbefehle], Seite 225, Abschnitt A.10.1 [Font], Seite 646.

Schriftarten für einen Eintrag
Jede Schriftart, die über das Betriebssystem installiert ist und von FontConfig erkannt wird, kann in einer Partitur eingefügt werden. Dazu verwendet man folgende Syntax:

\override Staff.TimeSignature.font-name = "Bitstream Charter"
\override Staff.TimeSignature.font-size = #2
\time 3/4

a'1\markup {
  \override #'{font-name . "Bitstream Vera Sans,sans-serif, Oblique Bold")
  { Vera Oblique Bold }
}

\begin{Verbatim}
\texttt{Vera Oblique Bold}
\end{Verbatim}

Mit folgendem Befehl erhält man eine Liste aller verfügbaren Schriftarten des Betriebssystems:

lilypond -dshow-available-fonts x

Siehe auch
Notationsreferenz: [Was sind Schriftarten], Seite 237, [Schriftart des gesamten Dokuments], Seite 239.

Schriftart des gesamten Dokuments
Es ist auch möglich, die Schriftarten für die gesamte Partitur zu ändern. In diesem Fall müssen die Familien roman, sans und typewriter in genaus dieser Reihenfolge entsprechend der Syntax unten definiert werden. Einzelheiten zu Schriftarten in [Was sind Schriftarten], Seite 237.

\begin{Verbatim}
\texttt{paper {}
  myStaffSize = #20
  #(define fonts
    (make-pango-font-tree "Times New Roman"
\end{Verbatim}
Siehe auch

Notationsreferenz: [Was sind Schriftarten], Seite 237, [Schriftarten für einen Eintrag], Seite 239, [Überblick über die wichtigsten Textbeschriftungsbefehle], Seite 225, Abschnitt A.10.1 [Font], Seite 646.
2 Spezielle Notation

Dieser Abschnitt erklärt, wie Notation erstellt wird, die nur für ein bestimmtes Instrument oder einen Stil eingesetzt wird.

2.1 Notation von Gesang

Dieser Abschnitt erklärt, wie Vokalmusik gesetzt werden kann und die Silben von Gesangstext an den Noten ausgerichtet werden.

2.1.1 Übliche Notation für Vokalmusik

Dieser Abschnitt erklärt Eigenheiten und Probleme, die die meisten Arten an Vokalmusik gemeinsam haben.

Referenz für Vokalmusik

Dieser Abschnitt, wo man Lösungen zu den Problemen finden kann, die bei der Notation von Gesang mit Text auftreten können.

- Vokalmusik braucht oft die Benutzung von Textbeschriftung (dem markup-Modus) für den Gesangstext oder andere Textelemente (Namen von Figuren usw.). Die entsprechende Syntax ist beschrieben in [Textbeschriftung (Einleitung)], Seite 223.
- ’Ambitus’ können zu Beginn der Stimmen hinzugefügt werden, dies findet sich erklärt in [Tonumfang], Seite 31.
- Dynamikbezeichnung werden normalerweise unter das Notensystem platziert, aber in Chormusik werden sie normalerweise über das Notensystem notiert, um Platz für den Text zu schaffen. Siehe [Partiturbeispiele für Chormusik], Seite 280.
Siehe auch

Glossar: Abschnitt “ambitus” in Glossar.
Notationsreferenz: [Textbeschifung (Einleitung)], Seite 223, [Tonumfang], Seite 31, [Partiturbeispiele für Chormusik], Seite 280.

Eingabe von Text

Gesangstext muss in einem speziellen Modus notiert werden. Der Gesangstextmodus kann mit der Umgebung \lyricmode angegeben werden, oder indem \addlyrics bzw. \lyricsto eingesetzt wird. In diesem Modus kann Text mit Akzenten und Satzzeichen notiert werden, und das Programm liest d nicht als die Tonhöhe D, sondern als eine Silbe Text. Anders gesagt: Silben werden wie Noten notiert, aber die Tonhöhen werden durch Text ersetzt.

Beispielsweise:
\lyricmode { Gern4 hätt'4 ich4 dich4 lieb!2 }

Es gibt zwei generelle Methoden, die horizontale Orientierung der Textsilben anzugeben, entweder indem ihre Dauer angegeben wird, wie oben in dem Beispiel, oder indem die Silben automatisch an den Noten ausgerichtet werden. Dazu muss entweder \addlyrics oder \lyricsto eingesetzt werden. Die erste Methode ist beschrieben in [Manuelle Silbendauern], Seite 248, die zweite in [Automatische Silbendauern], Seite 245.


Jedes Zeichen, das nicht Leerzeichen noch Zahl ist, wird als Bestandteil der Silbe angesehen. Eine Silbe kann also auch mit } enden, was oft zu dem Fehler \lyricmode { lah- lah} führen kann. Hier wird } als Teil der letzten Silbe gerechnet, so dass die öffnende Klammer keine schließende Klammer hat und die Eingabedatei nicht funktioniert. Klammern sollten deshalb immer von Leerzeichen umgeben sein.

\lyricmode { lah lah lah }

Auch ein Punkt, der auf eine Silbe folgt, wird in die Silbe inkorporiert. Infolgedessen müssen auch um Eigenschaftsbezeichnungen Leerzeichen gesetzt werden. Ein Befehl heißt also nicht:
\override Score.LyricText.font-shape = #'italic

sondern
\override Score.LyricText.font-shape = #’italic

Punkte, Gesangstext mit Akzenten, verschiedene lateinische und nicht-lateinische Zeichen sowie auch etwa Sonderzeichen (wie ein Herz-Symbol) können direkt in die Notationsdatei geschrieben werden. Es muss dabei sichergestellt werden, dass die Datei in der UTF-8-Kodierung gespeichert wird. Zu mehr Information siehe Abschnitt 3.3.3 [Sonderzeichen], Seite 482.

\relative { d'8 c16 a bes8 f e' d c4 }
\addlyrics { „Schad’ um das schö -- ne grü -- ne Band, }

„Schad’ um das schöne grüne Band,
Kapitel 2: Spezielle Notation

Normale Anführungszeichen können im Gesangstext auch benutzt werden, aber sie müssen mit einem Backslash und weiteren Anführungszeichen begleitet werden:

\relative { \time 3/4 e'4 e4. e8 d4 e d c2. }
\addlyrics { ""I" am so lone -- "ly,\"" said she }


Man hat sehr gut Möglichkeiten, die Erscheinung des Gesangstextes zu beeinflussen, wenn man dafür Textbeschriftungsbefehle einsetzt. Siehe hierzu Abschnitt 1.8.2 [Text formatieren], Seite 223.

Ausgewählte Schnipsel

**Silben im Gesangstext formatieren**

Textbeschriftungsmodus kann eingesetzt werden, um individuelle Silben im Gesangstext zu formatieren.

mel = \relative c' { c4 c c c }
lyr = \lyricmode {
Lyrics \markup { \italic can } \markup { \with-color #red contain }
\markup { \fontsize #8 \bold Markup! }
}

<<
\new Voice = melody \mel
\new Lyrics \lyricsto melody \lyr
>>

Lyrics can contain **Markup**!

Siehe auch


Notationsreferenz: [Automatische Silbendauern], Seite 245, Abschnitt 1.8.3 [Schriftarten], Seite 237, Abschnitt 5.4.1 [Eingabe-Modi], Seite 580, [Manuelle Silbendauern], Seite 248, Abschnitt 1.8.2 [Text formatieren], Seite 223, Abschnitt 3.3.3 [Sonderzeichen], Seite 482.

Referenz der Interna: Abschnitt “LyricText” in Referenz der Interna.

Text an einer Melodie ausrichten

Gesangstext wird gesetzt, wenn er in einem \texttt{Lyrics}-Kontext ausgewertet wird, siehe Abschnitt 5.1.1 [Was sind Kontexte?], Seite 552.

\texttt{\new Lyrics \lyricmode \{ ... \}}

Gesangstext kann an einer Melodie auf zwei Arten ausgerichtet werden:

- Gesangstext kann automatisch ausgerichtet werden, wobei die Dauern von einer Notenstimme oder (in besonderen Umstän-\texttt{\addlyrics, \lyricsto} oder indem man die \texttt{associatedVoice}-Eigenschaft einsetzt. Mehr Informationen in [Automatische Silbendauern], Seite 245.

\begin{verbatim}
\new Staff <<
\time 2/4
\new Voice = "one" \relative {
  \voiceOne
c'4 b8. a16 g4. r8 a4 ( b ) c2
}
\new Voice = "two" \relative {
  \voiceTwo
s2 s4. f'8 e4 d c2
}
>>

% takes durations and alignment from notes in "one"
\new Lyrics \lyricsto "one" {
  Life is __ _ love, live __ life.
}

% takes durations and alignment from notes in "one" initially
% then switches to "two"
\new Lyrics \lyricsto "one" {
  No more let
  \set associatedVoice = "two" % must be set one syllable early
  sins and sor -- rows grow.
}
\end{verbatim}

Die erste Strophe zeigt, wie normalerweise Text gesetzt wird.

Die zweite Strophe zeigt, wie die Stimme (Voice), von der die Dauern entnommen werden, geändert werden kann. Das ist nützlich, wenn der Text unterschiedlicher Strophen die Noten auf unterschiedliche Weise füllt und alle Dauern als Voice-Kontexte vorhanden sind. Zu mehr Details siehe Abschnitt 2.1.3 [Strophen], Seite 271.

- Gesangstext kann unabhängig von den Notendauern platziert werden, indem man die Dauern der Silben explizit vorgibt und den Text innerhalb von \texttt{\lyricmode} notiert:

\begin{verbatim}
\new lyrics \lyricmode
\end{verbatim}
Die erste Strophe wird nicht an den Noten ausgerichtet, weil die Dauer nicht angegeben sind, und der erste Wert 2 für alle Silben benutzt wird.


Wenn Text auf diese Weise eingegeben wird, werden die Silben links an den Noten ausgerichtet, können aber auch zentriert werden, indem man eine assoziierte Stimme angibt, wenn eine existiert. Zu Einzelheiten siehe Manuelle Silbendauern, Seite 248.

**Siehe auch**


Notation Reference: Abschnitt 5.1.1 [Was sind Kontexte?], Seite 552, [Automatische Silbendauern], Seite 245, Abschnitt 2.1.3 [Strophen], Seite 271, Manuelle Silbendauern, Seite 248, [Dialog zur Musik], Seite 289.

Referenz der Interna: Abschnitt “Lyrics” in Referenz der Interna.

**Automatische Silbendauern**

Die Silben des Gesangstextes können automatisch an einer Melodie ausgerichtet werden. Das ist auf drei Arten möglich:

- indem man einen benannten Voice-Kontext mit der Melodie durch \lyricsto zuweist,
- indem man den Gesangstext mit \addlyrics beginnt und direkt nach dem Voice-Kontext mit der Melodie notiert,
In allen drei Methoden können Bindestriche zwischen den Silben oder Fülllinien hinter einem Wortende gezogen werden. Zu Einzelheiten siehe [Fülllinien und Trennstriche], Seite 253.

Der Voice-Kontext, der die Melodie enthält, an der der Text ausgerichtet werden soll, darf noch nicht „gestorben“ sein, weil sonst aller Text danach verloren geht. Das kann passieren, wenn es Strecken gibt, in denen die Stimme pausiert. Zu Methoden, wie man Kontexte am Leben erhält, siehe Abschnitt 5.1.3 [Kontexte am Leben halten], Seite 555.

\lyricsto Benutzen
Gesangstext kann an einer Melodie automatisch ausgerichtet werden, indem man den benannten Voice-Kontext mit der Melodie durch den Befehl \lyricsto angibt:

\lyricsto \new Voice = "melody" {
  a4 a a a
}\new Lyrics \lyricsto "melody" {
  These are the words
}

These are the words


\addlyrics benutzen
Der \addlyrics-Befehl ist eigentlich nur eine Abkürzung für eine etwas kompliziertere LilyPond-Struktur, den man manchmal aus Bequemlichkeit einsetzen kann.

\addlyrics { Noten }

bedeutet das Gleiche wie

\new Voice = "bla" { Noten }
\new Lyrics \lyricsto "bla" { Gesangstext }

Hier ein Beispiel:

\addlyrics { play the game }

play the game
Weitere Strophen können mit weiteren \addlyrics-Abschnitten hinzugefügt werden:

\begin{verbatim}
{ 
  \time 3/4 
  \relative { c'2 e4 g2. } 
  \addlyrics { play the game } 
  \addlyrics { speel het spel } 
  \addlyrics { joue le jeu } 
}
\end{verbatim}

Der Befehl \addlyrics kann keine polyphonen Situationen bewältigen. In diesen Fällen sollen man \lyricsto benutzen.

\textbf{associatedVoice benutzen}

Die Melodie, an die der Gesangstext ausgerichtet wird, kann durch Setzen der associatedVoice-Eigenschaft geändert werden:

\begin{verbatim}
\set associatedVoice = "lala"
\end{verbatim}

Der Wert der Eigenschaft (hier "lala") ist die Bezeichnung eines Voice-Kontextes. Aus technischen Gründen muss der \set-Befehl eine Silbe vor der Silbe gesetzt werden, auf die er wirken soll.

Ein Beispiel demonstriert das:

\begin{verbatim}
<< 
\new Staff << 
\time 2/4 
\new Voice = "one" \relative { 
  \voiceOne 
  c'4 b8. a16 g4. r8 a4 ( b ) c2 
} 
\new Voice = "two" \relative { 
  \voiceTwo 
  s2 s4. f'8 e8 d4. c2 
} 
% takes durations and alignment from notes in "one" initially 
% then switches to "two" 
\new Lyrics \lyricsto "one" { 
  No more let 
  \set associatedVoice = "two" % must be set one syllable early 
  sins and sor -- rows grow. 
} 
>>
\end{verbatim}
Siehe auch

Notationsreferenz: [Fülllinien und Trennstriche], Seite 253, Abschnitt 5.1.3 [Kontexte am Leben halten], Seite 555, [Gesangstext vertikal verschieben], Seite 255.

Manuelle Silbendauern


Hier zwei Beispiele:

```
<<
\new Voice = "melody" {
    \time 3/4
    c2 e4 g2 f
}
\new Lyrics \lyricmode {
    play1 the4 game4
}
>>
```

```
<<
\new Staff {
    \relative {
        c'2 c2
d1
    }
}
\new Lyrics {
    \lyricmode {
        I2 like4. my8 cat!1
    }
}
\new Staff {
    \relative {
        c'8 c c c c c c c
c8 c c c c c c c
    }
}
```
Diese Technik ist nützlich, wenn man Dialog zur Musik schreiben will, siehe [Dialog zur Musik], Seite 289.

Um Silben an den Noten des entsprechenden musikalischen Moments zu zentrieren, muss \associatedVoice auf die Bezeichnung des Stimmen-Kontext eingestellt werden, in dem sich die Noten befinden. Wenn \associatedVoice definiert ist, können doppelte Bindestriche zwischen Silben und doppelte Unterstriche hinter Wörtern für Fülllinien benutzt werden:

\begin{verbatim}
<<
    \new Voice = "melody" {
        \time 3/4
        c2 e4 g f g
    }
    \new Lyrics \lyricmode {
        \set \associatedVoice = "melody"
        play2 the4 game2. __
    }
>>
\end{verbatim}

Siehe auch

Notationsreferenz: [Dialog zur Musik], Seite 289, Abschnitt 5.1.3 [Kontexte am Leben halten], Seite 555.


Mehrere Silben zu einer Note

Um mehr als eine Silbe zu einer Note zuzuordnen, können die Silben mit geraden Anführungszeichen (") umgeben werden oder ein Unterstrich (_) benutzt werden, um ein Leerzeichen zwischen Silben zu setzen. Mit der Tilde (~) kann ein Bindebogen gesetzt werden.

\begin{verbatim}
{ 
    \relative { \autoBeamOff 
        r8 b' c fis, fis c' b e, } 
    \addlyrics { Che\_in ques -- ta\_e\_in quel -- l'al -- tr'on -- da } 
    \addlyrics { "Che\_in" ques -- "ta e\_in" quel -- l'al -- tr'on -- da } 
    \addlyrics { Che~in ques -- ta~e~in quel -- l'al -- tr'on -- da } 
} 
\end{verbatim}
Mehrere Noten zu einer Silbe

Öfters, insbesondere in Alter Musik, wird eine einzige Silbe zu mehreren Noten gesungen, was als Melisma bezeichnet wird. Die Silbe eines Melismas wird normalerweise links an der ersten Note des Melismas ausgerichtet.

Melismen können direkt im Gesangstext definiert werden, indem ein Unterstrich (_) für jede Note notiert wird, die übersprungen werden soll.

Wenn ein Melisma an einer Silbe auftritt, die nicht die letzte eines Wortes ist, wird diese Silbe mit der folgenden durch wiederholte Trennstriche verbunden. Dies wird notiert, indem man zwei Trennstriche (--) nach der Silbe notiert.

Wenn ein Melisma an der letzten Silbe eines Wortes auftritt, wird eine Fülllinie vom Ende der Silbe bis zur letzten Note des Melismas gezeichnet. Das wird durch zwei Unterstriche (__) nach der Silbe notiert.

Es gibt fünf Arten, auf die ein Melisma angezeigt werden kann:

- Melismen werden automatisch zu Noten erstellt, die übergebunden sind:

\begin{verbatim}
\new Voice = "melody" \relative {
  \time 3/4
  f'4 g2 ~ |
  4 e2 ~ |
  8 }
\new Lyrics \lyricsto "melody" { Ky -- ri -- e__ }
\end{verbatim}

- Melismen können automatisch aus den Noten erstellt werden, indem man Legatobögen über den Noten eines Melismas notiert. Auf diese Weise wird Gesangstext üblicherweise notiert:

\begin{verbatim}
\new Voice = "melody" \relative {
  \time 3/4
  f'4 g8 ( f e f )
  e8 ( d e2 )
}
\new Lyrics \lyricsto "melody" { Ky -- ri -- e__ }
\end{verbatim}
Dabei ist zu beachten, dass Phrasierungsbogen die Erstellung eines Melismas nicht beeinflussen.

- Noten werden als ein Melisma betrachtet, wenn sie manuell mit einem Balken versehen werden, vorausgesetzt, dass die automatische Bebalkung ausgeschaltet ist. Siehe [Einstellung von automatischen Balken], Seite 80.

Diese Methode eignet sich natürlich nicht für Noten, die länger als Achtel sind.

- Eine Gruppe von Noten ohne Legatobogen werden als Melisma betrachtet, wenn sie zwischen \melisma und \melismaEnd eingeschlossen sind:
Ein Melisma kann auch ausschließlich im Gesangstext notiert werden, indem man einzelnene Unterstriche (_) für jede Note eingibt, die zum Melisma hinzugefügt werden soll.

\begin{verbatim}
\new Voice = "melody" \relative {
  \time 3/4
  f''4 g8 f e f
  e8 d e2}
\end{verbatim}

Man kann durchaus auch Binde- und Legatobögen sowie manuelle Balken benutzen, ohne dass sie Melismen bezeichnen, wenn \texttt{melismaBusyProperties} aufgerufen wird:

\begin{verbatim}
\new Voice = "melody" \relative {
  \time 3/4
  \set melismaBusyProperties = #'() c'4 d ( e )
  g8 [ f ] f4 ~ 4}
\end{verbatim}

Andere Einstellungen für \texttt{melismaBusyProperties} können beutzt werden, um nur eine Auswahl von Binde-, Legatobögen und Balken zur automatischen Melismenerkennung gelten zu lassen. Siehe \texttt{melismaBusyProperties} in Abschnitt “Tunable context properties” in Referenz der Interna.

Alternativ kann auch \texttt{ignoreMelismata} auf wahr gesetzt werden, wenn alle Melisma-Bezeichnungen ignoriert werden sollen, siehe [Strophen mit unterschiedlichem Rhythmus], Seite 272.

Wenn ein Melisma während einer Passage benötigt wird, in der \texttt{melismaBusyProperties} aktiviert ist, kann es durch einen einzelnen Unterstrich im Gesangstext für jede Note des Melismas angegeben werden:

\begin{verbatim}
\new Voice = "melody" \relative {
  \time 3/4
  \set melismaBusyProperties = #'()}
\end{verbatim}
Kapitel 2: Spezielle Notation

Vordefinierte Befehle
\autoBeamOff, \autoBeamOn, \melisma, \melismaEnd.

Siehe auch
Glossar: Abschnitt “melisma” in Glossar.
Notation Reference: Abschnitt 5.1.3 [Kontexte am Leben halten], Seite 555, [Automatische Silbendauern], Seite 245, [Einstellung von automatischen Balken], Seite 80, [Strophen mit unterschiedlichem Rhythmus], Seite 272.
Internals Reference: Abschnitt “Tunable context properties” in Referenz der Interna.

Bekannte Probleme und Warnungen
Fülllinien zu Melismen werden nicht automatisch erkannt, sondern müssen manuell gesetzt werden.

Fülllinien und Trennstriche
Wenn die letzte Silbe eines Wortes auf ein Melisma fällt, wird das Melisma oft mit einer langen horizontalen Linie angezeigt, die nach dem Wort beginnt und mit der letzten Note des Melismas endet. Derartige Fülllinien werden mit einem doppelten Unterstrich (_ _) eingeben, wobei beachtet werden muss, dass er von Leerzeichen umgeben ist.

Zentrierte Bindestriche zwischen den einzelnen Silben werden mit einem doppelten Bindestrich ( _ _ ) eingeben, wobei beachtet werden muss, dass er von Leerzeichen umgeben ist. Der Bindestrich wird zwischen den Silben zentriert und seine Länge dem Notenabstand angepasst.

In sehr eng notierter Musik können die Bindestriche ganz wegfallen. Dieses Verhalten kann aber auch unterbunden werden, wenn den Eigenschaften minimum-distance (minimaler Abstand zwischen Silben) und minimum-length (Wert, unterhalb von dem Bindestriche wegfallen) andere Werte erhalten. Beide sind Eigenschaften von LyricHyphen.

Siehe auch
Referenz der Interna: Abschnitt “LyricExtender” in Referenz der Interna, Abschnitt “LyricHyphen” in Referenz der Interna.

\new Lyrics \lyricsto "melody" { Ky -- ri -- e __ _ _ _ _ _ _ _ _ _ _ }
2.1.2 Techniken für die Gesangstextnotation

Mit Gesangstexten und Bezeichnern arbeiten

Um Variablen zu definieren, die Gesangstext beinhalten, muss die \lyricmode-Umgebung benutzt werden.

```
musicOne = \relative { 
c''4 b8. a16 g4. f8 e4 d c2 
} 
verseOne = \lyricmode { 
    Joy to the world, the Lord is come. 
} 
\score { 
<< 
\new Voice = "one" { 
    \time 2/4 
    \musicOne 
} 
\new Lyrics \lyricsto "one" { 
    \verseOne 
} 
>> 
}
```

Dauern müssen nicht angegeben werden, wenn die Variable im Zusammenhang mit \addlyrics oder \lyricsto aufgerufen wird.

Für eine andere Anordnung oder kompliziertere Situationen bietet es sich an, zuerst Systeme und Gesangstextumgebungen zu definieren, dann die Hierarchie von Systemen und Gesangstextzeilen aufzustellen ohne den Gesangstext selber aufzurufen, und dann den Gesangstext mit \context darunter aufzurufen. Das stellt sicher, dass die Stimmen, die durch \lyricsto angefordert werden, auch immer schon definiert sind. Beispielsweise:

```
sopranoMusic = \relative { c''4 c c c } 
contraltoMusic = \relative { a'4 a a a } 
sopranoWords = \lyricmode { Sop -- ra -- no words } 
contraltoWords = \lyricmode { Con -- tral -- to words } 
\score { 
\new ChoirStaff << 
\new Staff { 
    \new Voice = "sopranos" { 
        \sopranoMusic 
    } 
} 
\new Lyrics = "sopranos" 
\new Lyrics = "contraltos" 
\new Staff { 
    \new Voice = "contraltos" 
}
```
Abhängig von der Art der Musik kann der Gesangstext über oder unter einem Notensystem oder zwischen zwei Systemen positioniert werden. Es ist am einfachsten, den Text unter das verknüpfte System zu positionieren, was man erreicht, indem man den Lyrics-Kontext direkt unter dem System definiert:

```
\score {
    \new Staff {
        \new Voice = "melody" {
            \relative { c'4 c c c }
        }
    }
    \new Lyrics {
        \lyricsto "melody" {
            Here are the words
        }
    }
} >>
```
Alternativ kann auch ein zweistufiger Prozess benutzt werden. Zuerst wird der Gesangstextkontext definiert (ohne jeglichen Inhalt), bevor Stimm- und Systemkontexte definiert wurden. Dann wird der \lyricsto-Befehl nach der Definition der Stimme, auf die er verweist, mit \context angegeben:

\score {
<<
\new Lyrics = "lyrics" \with {
% lyrics above a staff should have this override
\override VerticalAxisGroup.staff-affinity = #DOWN
}
\new Staff {
\new Voice = "melody" {
\relative { c''4 c c c }
}
}
\context Lyrics = "lyrics" {
\lyricsto "melody" {
Here are the words
}
}
>>
}
Wenn zwei Stimmen sich auf unterschiedlichen Systemen befinden, kann der Text zwischen den Systemen platziert werden, wobei beide der Methoden eines etzt werden können. Hier ein Beispiel für die zweite Methode:

```latex
\score {
  \new ChoirStaff <<
  \new Staff {
    \new Voice = "sopranos" {
      \relative { c''4 c c c }
    }
  }
  \new Lyrics = "sopranos"
  \new Lyrics = "contraltos" \with {
    % lyrics above a staff should have this override
    \override VerticalAxisGroup.staff-affinity = #DOWN
  }
  \new Staff {
    \new Voice = "contraltos" {
      \relative { a'4 a a a }
    }
  }
  \context Lyrics = "sopranos" {
    \lyricsto "sopranos" {
      Sop -- ra -- no words
    }
  }
  \context Lyrics = "contraltos" {
    \lyricsto "contraltos" {
      Con -- tral -- to words
    }
  }
  >>
}
```

Andere Kombinationen von Gesangstext und System können erstellt werden, indem man die gegebenen Beispiele modifiziert oder auch die Abschnitt "Vokalensemble" in *Handbuch zum Lernen*-Vorlagen im Handbuch zum Lernen heranzieht.

**Ausgewählte Schnipsel**

*Platzverteilung von Gesangstext wie in 2.12 für neuere Versionen*


```latex
global = {
```
\key d \major
\time 3/4
}

sopMusic = \relative c' {
  \% VERSE ONE
  fis4 fis fis | \break
  fis4. e8 e4
}

altoMusic = \relative c' {
  \% VERSE ONE
  d4 d d |
  d4. b8 b4 |
}

tenorMusic = \relative c' {
  a4 a a |
  b4. g8 g4 |
}

bassMusic = \relative c {
  d4 d d |
  g,4. g8 g4 |
}

words = \lyricmode {
  Great is Thy faith -- ful -- ness,
}

\score {
  \new ChoirStaff <<
    \new Lyrics = sopranos
    \new Staff = women <<
      \new Voice = "sopranos" {
        \voiceOne
        \global \sopMusic
      }
      \new Voice = "altos" {
        \voiceTwo
        \global \altoMusic
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors"
    \new Staff = men <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        \global \tenorMusic
      }
      \new Voice = "basses" {

\voiceTwo \global \bassMusic
}

\new Lyrics = basses
\context Lyrics = sopranos \lyricsto sopranos \words
\context Lyrics = altos \lyricsto altos \words
\context Lyrics = tenors \lyricsto tenors \words
\context Lyrics = basses \lyricsto basses \words

\layout {
\context {
\Lyrics
\override VerticalAxisGroup.staff-affinity = ##f
\override VerticalAxisGroup.staff-staff-spacing = #'((basic-distance . 0)
  (minimum-distance . 2)
  (padding . 2))
}
\context {
\Staff
\override VerticalAxisGroup.staff-staff-spacing = #'((basic-distance . 0)
  (minimum-distance . 2)
  (padding . 2))
}
}
}
Siehe auch


Notation Reference: Abschnitt 5.1.7 [Reihenfolge des Kontextlayouts], Seite 567, Abschnitt 5.1.2 [Kontexte erstellen und referenzieren], Seite 554.

Silben horizontal verschieben

Um den Abstand zwischen Silben zu vergrößern, kann die minimum-distance-Eigenschaft des LyricSpace-Objekts gesetzt werden:

\relative c' {
  c c c c
  \override Lyrics.LyricSpace.minimum-distance = #1.0
  c c c c
}

\addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
}

Damit diese Einstellung für alle Gesangtextzeilen in einer Partitur wirkt, muss sie im layout-Block vorgenommen werden.

\score {
  \relative {
    c' c c c
    c c c c
  }
  \addlyrics {
    longtext longtext longtext longtext
    longtext longtext longtext longtext
  }
}
Ausgewählte Schnipsel

Ausrichtung von Gesangstext

Die horizontale Ausrichtung von Gesangstext kann eingestellt werden, indem man die self-alignment-X-Eigenschaft des LyricText-Objekts verändert. \#-1 bedeutet links, \#0 bedeutet mittig und \#1 bedeutet rechts, man kann aber genauso gut auch \#LEFT, \#CENTER und \#RIGHT benutzen.

\layout { ragged-right = ##f }
\relative c'' { c1 c1 c1 }
\addlyrics { \once \override LyricText.self-alignment-X = \#LEFT "This is left-aligned"
    \once \override LyricText.self-alignment-X = \#CENTER "This is centered"
    \once \override LyricText.self-alignment-X = \#1 "This is right-aligned"
}

This is left-aligned This is centered This is right-aligned

Eine Überprüfung, mit der sichergestellt wird, dass kein Text in die Seitenränder ragt, verlangt zusätzliche Computerzeit. Um den Notensatz etwas zu beschleunigen, kann die Überprüfung abgestellt werden:

\override Score.PaperColumn.keep-inside-line = ##f
Kapitel 2: Spezielle Notation

Damit Gesangstext auch nicht mit Taktlinien zusammenstößt, kann folgende Einstellung gesetzt werden:

\layout {
\context {
\Lyrics
\consists "Bar_engraver"
\consists "Separating_line_group_engraver"
hide BarLine
}
}

Gesangstext und Wiederholungen

Einfache Wiederholungen

Wiederholungen von Musik ist vollständig an anderer Stelle behandelt, siehe Abschnitt 1.4 [Wiederholungszeichen], Seite 136. Dieser Abschnitt erklärt, wie man Gesangstext zu wiederholten Noten hinzufügt.

Gesangstext zu einem Abschnitt, der wiederholt wird, muss in der gleichen Wiederholungskonstruktion wie die Noten enthalten sein, wenn der Text sich nicht ändert:

\score {
<<
\new Staff {
\new Voice = "melody" {
\relative {
   a'4 a a a
   \repeat volta 2 { b4 b b b }
}
}
\new Lyrics {
\lyricsto "melody" {
Not re -- peat -- ed.
\repeat volta 2 { Re -- peat -- ed twice. }
}
}
>>
}

Der Text wird dann richtig erweitert, wenn die Wiederholung mit \unfoldRepeats aus-geklappt wird:

\score {
\unfoldRepeats {
\new Staff {
\new Voice = "melody" {
\relative {

Kapitel 2: Spezielle Notation

Wenn der wiederholte Abschnitt unterschiedlichen Text hat und ausgeklappt werden soll, müssen alle Wörter notiert werden:

\score {
  \new Staff {
    \new Voice = "melody" {
      \relative {
        a'4 a a a
        \repeat unfold 2 { b4 b b b }
      }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      Not re--peat--ed.
      \repeat volta 2 { Re--peat--ed twice. }
    }
  }
  >>
}

\[f clef \times\frac{C4}{4}\]


Wenn der Text in einer Wiederholung mit volta (also mit punktiertem Doppelstrich) unterschiedlich ist, muss der Text jeder Wiederholung in einem eigenen Lyrics-Kontext notiert werden, der korrekt in parallelen Abschnitten geschachtelt wird:

\score {
  \new Staff {
    \new Voice = "melody" {
      \relative {
        a'4 a a a
        \repeat unfold 2 { b4 b b b }
      }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      Not re--peat--ed.
      The first time words.
      Sec--ond time words.
    }
  }
  >>
}

\[f clef \times\frac{C4}{4}\]

Not repeated. The first time words. Second time words.
\new Staff {
  \new Voice = "melody" {
    \relative {
      a'4 a a a
      \repeat volta 2 { b4 b b b }
    }
  }
}
\new Lyrics \lyricsto "melody" {
  Not re--peat--ed.
  <<
  \set associatedVoice = "melody"
  \{ The first time words. \}
  \new Lyrics {
    Sec -- ond time words.
  }
  >>
  >>
}

Not repeated. The first time words.
Sec--ond time words.

Neue Strophen können auf die gleiche Art hinzugefügt werden:
\score {
  <<
    \new Staff {
      \new Voice = "singleVoice" {
        \relative {
          a'4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
    \new Lyrics \lyricsto "singleVoice" {
      Not re--peat--ed.
      <<
      \set associatedVoice = "singleVoice"
      \{ The first time words. \}
      \new Lyrics {
        Sec -- ond time words.
      }
      \new Lyrics {
        \set associatedVoice = "singleVoice"
        The third time words.
      }
      >>
  }
Wenn diese Konstruktion jedoch innerhalb eines Mehrsystemkontexts eingebettet ist, wie etwa ein ChoirStaff, werden die Texte der zweiten und dritten Strophe unter dem untersten System ausgegeben werden.

Um sie richtig zu positionieren, kann \alignBelowContext eingesetzt werden:

\score {
<<
\new Staff {
  \new Voice = "melody" {
    \relative {
      a'4 a a a
      \repeat volta 3 { b4 b b b }
      c4 c c c
    }
  }
}
\new Lyrics = "firstVerse" \lyricsto "melody" {
  Not repeated.
  The first time words.
  The end section.
  Second time words.
  The third time words.
} <<
\new Lyrics = "secondVerse" \with { \alignBelowContext = "firstVerse" } {
  \set associatedVoice = "melody"
  Second time words.
} \new Lyrics = "thirdVerse" \with { \alignBelowContext = "secondVerse" } {
  \set associatedVoice = "melody"
  The third time words.
} >>
The end section.
\new Voice = "harmony" {
  \relative {
    f'4 f f f \repeat volta 2 { g8 g4 g2 } a4 a8. a16 a2
  }
} >>
}
Wiederholungen mit alternativen Endungen

Wenn der Text des wiederholten Abschnitts der gleiche ist, kann die gleiche Struktur für Gesangstext und Noten eingesetzt werden.

\score {
<<
\new Staff {
\time 2/4
\new Voice = "melody" {
\relative {
 a'4 a a a
\repeat volta 2 { b4 b }
\alternative { { b b } { b c } }
}
}
\new Lyrics {
\lyricsto "melody" {
Not re--peat--ed.
\repeat volta 2 { Re--peat-- } 
\alternative { { ed twice. } { ed twice. } }
}
}
>>
}

Aber wenn der wiederholte Abschnitt unterschiedlichen Text hat, kann keine repeat-Konstruktion um den Text eingesetzt werden und man muss manuell mit \skip-Befehlen die Noten überspringen, auf die sich der alternative Text nicht bezieht.

Achtung: hier darf kein Unterstrich (_) benutzt werden, um Noten zu überspringen, weil das ein Melisma anzeigen würde und die vorhergehende Silbe dazu veranlasst, links ausgerichtet zu werden.

\textit{Achtung:} Der \texttt{\skip}-Befehl muss von einer Zahl gefolgt werden, aber diese Zahl wird ignoriert, wenn der Gesangstext seine Dauern von einer assoziierten Melodie ableitet, die mit \texttt{\addlyrics} oder \texttt{\lyricsto} angefügt wird. Jeder \texttt{\skip}-Befehl überspringt eine einzelne Note beliebiger Dauer, unabhängig vom Wert der auf den Befehl folgenden Zahl.
Wenn eine Note zu zwei oder mehr alternativen Endungen übergebunden wird, wird ein Bindebogen benutzt, um die Note in die erste alternative Endung überzubinden, und ein \repeatTie benutzt, um in die zweite und alle weiteren Klammern zu überbinden. Diese Struktur erzeugt problematische Ausrichtungen, wenn ein Gesangstext hinzu kommt und verlängert die alternativen Klammern, sodass es besser sein kann, die übergebundenen Noten vollständig in die Klammern aufzunehmen, um ein besseres Resultat zu erhalten.

Der Bindebogen erstellt ein Melisma zur ersten Klammer, aber nicht zur zweiten und allen weiteren Klammern, sodass es nötig ist, die automatische Erstellung von Melismen für die Klammer-Abschnitte zu deaktivieren und manuell die Noten mit \skip zu überspringen, um eine richtige Ausrichtung des Textes zu erreichen.
Wenn \unfoldRepeats in einem Abschnitt eingesetzt wird, der den \repeatTie-Befehl enthält, sollte der \repeatTie entfernt werden, damit nicht beide Bindestriche ausgegeben werden.

Wenn der wiederholte Abschnitt unterschiedlichen Text hat, kann \repeat nicht um den Gesangstext benutzt werden, und \skip-Befehle müssen manuell eingegeben werden:

\score {
<<
\new Staff {
\time 2/4
\new Voice = "melody" {
\relative {
\repeat volta 2 { b'4 b ~}
\alternative { { b b } { b \repeatTie c } }
\unset melismaBusyProperties
c4 c
}
}
}\new Lyrics {
\lyricsto "melody" {
\repeat volta 2 { Here's a _ } 
\alternative {
{ \skip 1 verse }
{ \skip 1 sec }
}
ond one.
}
}
>>
\lyricsto "melody" {
\repeat volta 2 { Here's a __ } 
\alternative {
{ \skip 1 verse }
{ \skip 1 sec }
}
ond one.
}
\lyricsto "melody" {
Here's one
}
Wenn Sie Bindestriche und Fülllinien zwischen Wiederholung und Klammer benutzen wollen, müssen sie manuell notiert werden:

\score {
  \new Staff {
    \time 2/4
    \new Voice = "melody" {
      \relative {
        \repeat volta 2 { b'4 b ~}
        \alternative { { b b } { b \repeatTie c } }
        c4 c
      }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      Here's a __ verse.
      \repeat unfold 2 { \skip 1 }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      Here's "a_" sec -- ond one.
      \skip 1
    }
  }
}

Here's a__ verse.
Here's a__ second one.
Siehe auch

Notationsreferenz: Abschnitt 5.1.3 [Kontexte am Leben halten], Seite 555, Abschnitt 1.4 [Wiederholungszeichen], Seite 136.

Getrennte Texte

Wenn nur Text und Rhythmus von zwei Texten unterschiedlich sind, während die Noten gleich bleiben, kann man die automatische Melisma-Erkennung kurzzeitig ausschalten und das Melisma im Text anzeigen:

```
\score {  
  \new Voice = "melody" {  
    \relative c' {  
      \set melismaBusyProperties = #'()'  
      \slurDown  
      \slurDashed  
      e4 e8 ( e ) c4 c |  
      \unset melismaBusyProperties  
      c  
    }  
  }  
  \new Lyrics \lyricsto "melody" {  
    They shall not o\-ver\-come  
  }  
  \new Lyrics \lyricsto "melody" {  
    We will  
  }  
}  
```

Wenn sich sowohl Noten als auch Worte unterscheiden, kann es besser sein, die unterschiedlichen Noten und den Text zu notieren, indem man Voice-Kontexte benennt und den Text an die entsprechenden Kontexte anhängt:

```
\score {  
  \new Voice = "melody" {  
    \relative {  
      {  
        \voiceOne  
        e'4 e8 e  
      }  
    }  
  }  
  \new Voice = "splitpart" {  
    \voiceTwo  
    c4 c  
  }  
}  
```

\[\begin{align*}
\text{They shall not o\-ver\-come} \\
\text{We will}
\end{align*}\]
2.1.3 Strophen

Strophennummern hinzufügen

Strophennummerierung kann hinzugefügt werden:
\new Voice \relative { \time 3/4 g'2 e4 a2 f4 g2. } \addlyrics { \set stanza = "1. " Hi, my name is Bert. } \addlyrics { \set stanza = "2. " Oh, ché--ri, je t'aime }

Die Zahl wird direkt vor die erste Silbe gesetzt.

Lautstärkebezeichnung zu Strophen hinzufügen

Dynamikzeichen können zur Strophennummer hinzugefügt werden. In LilyPond muss alles, was vor einer Strophe gesetzt wird, im StanzaNumber-Objekt definiert werden, also auch Dynamikbezeichnung. Aus technischen Gründen muss die Strophe außerhalb von lyricmode gesetzt werden:

text = { \set stanza = \markup { \dynamic "ff" "1. " } \lyricmode { 1. Hi, my name is Bert. 2. Oh, ché--ri, je t'aime

}}
Big bang
}
}

<<
\new Voice = "tune" {
 \time 3/4
 g'4 c'2
 }
\new Lyrics \lyricsto "tune" \text
>>

Sängernamen zu Strophen hinzufügen


\new Voice \relative {
 \time 3/4 g'2 e4 a2 f4 g2.
 } \addlyrics {
 \set vocalName = "Bert"
 Hi, my name is Bert.
 } \addlyrics {
 \set vocalName = "Ernie"
 Oh, chérô, je t'aime
 }

Strophen mit unterschiedlichem Rhythmus

Oft haben unterschiedliche Strophen eines Liedes leicht unterschiedliche Silbenzahlen und werden darum auf andere Art zur Melodie gesungen. Derartige Variationen können mit \lyricsto bewältigt werden.

Melismen ignorieren


<<
\relative \new Voice = "lahlah" {
 \set Staff.autoBeaming = ##f
 c'4
 \slurDotted

Hi, my name is Bert.
Oh, chérô, je t’aime
\new Lyrics \lyricsto "lahlah" {
  more slow -- ly
}\new Lyrics \lyricsto "lahlah" {
  go
  \set ignoreMelismata = ##t
  fas -- ter
  \unset ignoreMelismata
  still
}\new Lyrics \lyricsto melody {
  normal
  \set includeGraceNotes = ##t
  case,
  gra -- ce case,
  after -- grace case,
  \set ignoreMelismata = ##t
  app. case,
  acc. case.
}
Bekannte Probleme und Warnungen

Wie bei \associatedVoice muss \includeGraceNotes spätestens eine Silbe vor derjenigen gesetzt werden, die unter einer Verzierungsnote stehen soll. Im Fall, dass eine Verzierungsnote die erste des Musikstückes ist, kann ein \with- oder \context-Block verwendet werden:

\begin{verbatim}
<< \new Voice = melody \relative c' { \grace { c16( d e f } g1) f } \new Lyrics \with { \includeGraceNotes = ##t } \lyricsto melody { Ah __ fa } >>
\end{verbatim}

\begin{center}
\includegraphics[width=0.5\textwidth]{example.png}
\end{center}

Zu einer alternativen Melodie umschalten

Es ist auch möglich, die Silben von verschiedenen Textzeilen an unterschiedlichen Melodien auszurichten. Das wird mit der \associatedVoice-Eigenschaft vorgenommen:

\begin{verbatim}
<< \relative \new Voice = "lahlah" { \set Staff.autoBeaming = ##f c'4 << \new Voice = "alternative" { \voiceOne \tuplet 3/2 { % show associations clearly. \override NoteColumn.force-hshift = #-3 f8 f g } } { \voiceTwo f8.[ g16] \oneVoice } >> a8( b) c } \new Lyrics \lyricsto "lahlah" { Ju -- ras -- sic Park } \new Lyrics \lyricsto "lahlah" { % Tricky: need to set \associatedVoice
\end{verbatim}
Kapitel 2: Spezielle Notation

% one syllable too soon!
\set associatedVoice = "alternative" % applies to "ran"
Ty --
ran --
no --
\set associatedVoice = "lahlah" % applies to "rus"
sau -- rus Rex
}

Der Text der ersten Strophe wird an der Stimme „lahlah“ ausgerichtet, aber die zweite Strophe wird zuerst zum lahhah-Kontext gesetzt und dann zur alternative-Melodie für die Silben „ran“ bis „sau“:

\set associatedVoice = "alternative" % applies to "ran"
Ty --
ran --
no --
\set associatedVoice = "lahlah" % applies to "rus"
sau -- rus Rex

Hier ist alternative die Bezeichnung des Voice-Kontexts mit der Triole.

Der \set associatedVoice-Befehl tritt eine Silbe zu früh auf, aber das ist in diesem Fall richtig.


Die Strophen am Ende ausdrucken

Manchmal soll nur eine Strophe mit der Melodie gesetzt werden und die weiteren Strophen als Text unter den Noten hinzugefügt werden. Dazu wird der Text in einer markup-Umgebung außerhalb der \score-Umgebung gesetzt. Es gibt zwei Arten, die Zeilen auszurichten, wie das Beispiel zeigt:

```
melody = \relative {
e' d c d | e e e e |
d d e d | c1 |
}

text = \lyricmode {
\set stanza = "1." Ma- ry had a lit- tle lamb,
its fleece was white as snow.
}
```

\score{ <<


\new Voice = "one" { \melody \\
new Lyrics \lyricsto "one" \text \\
>> \\
\layout { } \\
} \\
\markup { \column{
\line{ Verse 2. } \\
\line{ All the children laughed and played } \\
\line{ To see a lamb at school. } \\
} \\
} \\
\markup{
\wordwrap-string " \\
Verse 3.

Mary took it home again, 

It was against the rule." \\
}

1. Mary had a little lamb, its fleece was white as snow.

Verse 2. 
All the children laughed and played 
To see a lamb at school.

Verse 3. 
Mary took it home again, 
It was against the rule.

Die Strophen am Ende in mehreren Spalten drucken

\begin{verbatim}

melody = \relative {
  c'4 c c c | d d d d 
}

\set stanza = "1." This is verse one. 
It has two lines.

\end{verbatim}

\score{ << \\
\new Voice = "one" { \melody \\
\new Lyrics \lyricsto "one" \text \\
>>

\end{verbatim}

\end{verbatim}
277

1. This is verse one. It has two lines.

2. This is verse two. It has two lines.

3. This is verse three. It has two lines.

4. This is verse four. It has two lines.

5. This is verse five. It has two lines.
2.1.4 Lieder

Verweise für Lieder


Alle Notationselemente, die für die Notation von Liedern benötigt werden, werden woanders beschrieben:

- Um das Systemlayout einzurichten, siehe Abschnitt 1.6.1 [Systeme anzeigen lassen], Seite 174.
- Zur Notation von Klaviermusik siehe Abschnitt 2.2 [Tasteninstrumente und andere Instrumente mit mehreren Systemen], Seite 303.
- Zur Notation von Gesangstext zu einer Notenzeile siehe Abschnitt 2.1.1 [Übliche Notation für Vokalmusik], Seite 241.
- Zur Platzierung des Gesangstext siehe [Gesangstext vertikal verschieben], Seite 255.
- Zur Notation von Strophen siehe Abschnitt 2.1.3 [Strophen], Seite 271.
- Lieder werden oft auch mit Akkorden gesetzt, die als Symbole über dem Notensystem notiert werden. Das wird in Abschnitt 2.7.2 [Akkorde anzeigen], Seite 393, beschrieben.
- Zur Notation von Bunddiagrammen für die Akkorde einer Gitarrenbegleitung oder anderer Bundinstrumente siehe „Bund-Diagramm-Beschriftung“ in Abschnitt 2.4.1 [Übliche Notation für Saiteninstrumente mit Bünden], Seite 320.

Siehe auch


Notationsreferenz: Abschnitt 2.1.1 [Übliche Notation für Vokalmusik], Seite 241, Abschnitt 2.7.2 [Akkorde anzeigen], Seite 393, Abschnitt 1.6.1 [Systeme anzeigen lassen], Seite 174, Abschnitt 2.2 [Tasteninstrumente und andere Instrumente mit mehreren Systemen], Seite 303, Gesangstext vertikal verschieben, Seite 255, Abschnitt 2.1.3 [Strophen], Seite 271.


Liedblätter

Liedblätter können erstellt werden, indem man Gesangstext mit Akkorden im Akkord-Modus kombiniert; die Syntax ist erklärt in Abschnitt 2.7 [Notation von Akkorden], Seite 387.

Ausgewählte Schnipsel

Ein einfaches Liedblatt

Ein Liedblatt besteht aus Akkordbezeichnungen, einer Melodie und dem Liedtext:

```plaintext
<<
\chords { c2 g:sus4 f e }
\relative c' { 
a4 e c8 e r4 
b2 c4( d)
}```
Siëhe auch

Notationsreferenz: Abschnitt 2.7 [Notation von Akkorden], Seite 387.

2.1.5 Chormusik

Dieser Abschnitt zeigt Eigenheiten der Notation von Chormusik. Hierzu gehören Hymnen, mehrstimmige Lieder, Oratorien, Kantaten usw.

Verweise für Chormusik

Chormusik wird normalerweise auf zwei, drei oder vier Systemen innerhalb einer ChoirStaff-Gruppe notiert. Begleitung wird darunter als PianoStaff-Klaviersonsystem gesetzt, oft auch in kleinerer Größe, wenn es sich um eine Übungshilfe für ein a capella-Chorwerk handelt. Die Noten jeder Stimme werden in einem Voice-Kontext notiert, welche entweder einzeln auf einem eigenen Notensystem notiert werden oder zu zweit auf dem gleichen System gesetzt werden.


Einige häufig anzutreffende Sachverhalte für Chormusik sind woanders behandelt:

- Eine Einleitung, um SATB-Chorpartituren zu erstellen, findet sich in Abschnitt “Vierstimmige SATB-Partitur” in Handbuch zum Lernen.
- Einige Vorlagen, die sich für unterschiedliche Chormusik eignen, finden sich im Handbuch zum Lernen, siehe Abschnitt “Vokalensemble” in Handbuch zum Lernen.
- Zu Information über ChoirStaff und PianoStaff siehe [Systeme gruppieren], Seite 175.
- Besondere Notenköpfe, wie etwa für die „Sacred Harp“-Notation benutzt, finden sich in [Notenköpfe mit besonderen Formen], Seite 37.
- Wenn zwei Vokalstimmen sich ein System teilen, werden Hülse, Bögen usw. der oberen Stimme nach oben gerichtet und die der unteren Stimme nach unten. Dieses Verhalten erreicht man mit \voiceOne und \voiceTwo. Siehe [Mehrstimmigkeit in einem System], Seite 157.

Vordefinierte Befehle
\oneVoice, \voiceOne, \voiceTwo.

Siëhe auch


Notationsreferenz: [Systeme gruppieren], Seite 175, Abschnitt 5.1.7 [Reihenfolge des Kontextlayouts], Seite 567, [Notenköpfe mit besonderen Formen], Seite 37, [Mehrstimmigkeit in einem System], Seite 157.

Referenz der Interna: Abschnitt “ChoirStaff” in Referenz der Interna, Abschnitt “Lyrics” in Referenz der Interna, Abschnitt “PianoStaff” in Referenz der Interna.
Partiturbeispiele für Chormusik

Chormusik auf vier Systemen, mit oder ohne Klavierbegleitung, wird meistens mit zwei Systemgruppen pro Seite gesetzt. Abhängig von der Seitengröße kann das erfordern, dass die Standardgröße der Notensysteme geändert wird. Die folgenden Einstellungen sollten in Betracht gezogen werden:

- Die globale Systemgröße kann verändert werden, um die Größe aller Elemente einer Partitur zu ändern. Siehe Abschnitt 4.2.2 [Die Notensystemgröße einstellen], Seite 511.
- Die Dimensionen der vertikalen Layout-Variable können angezeigt werden, um beim Anpassen der vertikalen Platzverteilung zu helfen. Das und andere Möglichkeiten, die Noten auf weniger Seiten zu zwingen, finden sich in Abschnitt 4.6 [Die Musik auf weniger Seiten zwingen], Seite 548.
- Wenn die Anzahl der Systemgruppen pro Seite zwischen einer und mehreren wechselt, wird dies üblicherweise mit einem Trennzeichen zwischen den Systemgruppen angezeigt. Siehe [Systeme trennen], Seite 180.
- Zu Details für andere Eigenschaften der Seitenformatierung siehe Abschnitt 4.1 [Seitenlayout], Seite 498.

Dynamikzeichen werden in den Grundeinstellungen unter dem System notiert, aber in Chormusik werden sie oft über dem System gesetzt um nicht mit dem Gesangstext zu kollidieren. Der vordefiniert Befehl \dynamicUp erledigt das für einen Voice-Kontext auf einem eigenen Notensystem. Wenn mehrere Voice-Kontexte vorhanden sind, müsste man den Befehl in jedem einzeln setzen. Um alle Dynamikzeichen in einer Partitur über den Systemen zu setzen, kann eine erweiterte Form genutzt werden, wie das Beispiel zeigt:

\score {
    \new ChoirStaff <<
    \new Staff { 
        \new Voice {
            \relative { g'4\f g g g }
            }
        }
    \new Staff {
        \new Voice {
            \relative { d'4 d d\p d }
            }
        }
    >>
    \layout {
        \context { 
            \Score 
            \override DynamicText.direction = #UP
            \override DynamicLineSpanner.direction = #UP
        }
    }
}
Vordefinierte Befehle

\dynamicUp.

Siehe auch

Notationsreferenz: Abschnitt 4.2.2 [Die Notensystemgröße einstellen], Seite 511, Abschnitt 4.4 [Vertikale Abstände], Seite 520, Abschnitt 4.6.1 [Abstände anzeigen lassen], Seite 549, Abschnitt 4.6.2 [Abstände verändern], Seite 550, Abschnitt 4.2 [Partiturlayout], Seite 509, Abschnitt 4.3.8 [Eine zusätzliche Stimme für Umbrüche benutzen], Seite 518, Abschnitt 4.1 [Seitenlayout], Seite 498, [Systeme trennen], Seite 180, Abschnitt 4.6 [Die Musik auf weniger Seiten zwingen], Seite 548.


Geteilte Stimmen

Arpeggio-Klammern benutzen, um geteilte Stimmen besser sichtbar zu machen

Das arpeggioBracket-Klammereobjekt kann benutzt werden, um geteilte Stimmen anzuzeigen, wenn keine Hülse diese Information bieten. Das sieht man oft in Chormusik.

\include "english.ly"

\score {
    \relative c' {
        \key a \major 
        \time 2/2 
        <<
            \new Voice = "upper"
            <<
                { \voiceOne \arpeggioBracket
                    a2( b2
                    <\ b\ d>\ arpeggio)
                    <\ c\ s\ e>\ arpeggio ~
                    <\ c\ s\ e>4
                }
                \addlyrics { \lyricmode { A -- men. } }
            >>
            \new Voice = "lower"
            { \voiceTwo
                a1 ~
                a
                a ~
                a4 \bar "."
            }
            >>
        }
    } 
    \layout { ragged-right = ##t }
}
Siehe auch

Notationsreferenz: Abschnitt 1.3.3 [Ausdrucksbezeichnungen als Linien], Seite 129.

2.1.6 Oper und Musical

Noten, Text und Dialoge von Oper und Singspielen werden normalerweise auf eine der folgenden Weisen notiert:

- Eine *Aufführungspartitur* enthält alle Orchester- und Gesangsstimmen sowie Libretto-Stichworte der gesprochenen Abschnitte.
- *Orchesterstimmen* enthalten die Noten für einzelne Instrumente des Orchesters oder der Band.
- Ein *Vocal Book* enthält nur die Gesangsstimmen (ohne Begleitung), teilweise zusammen mit dem Libretto.
- Ein *Libretto* enthält die ausführlichen gesprochenen Abschnitte, wie man sie oft in Musicals oder Operetten findet, sowie den gesungenen Text. Normalerweise sind auch Regieanweisungen enthalten. LilyPond kann eingesetzt werden, um Libretti zu setzen, aber es kann günstiger sein, dafür eine andere Methode zu benutzen, da sie keine Noten enthalten.

Die Abschnitte der LilyPond-Dokumentation, die nützlich zum Setzen von Oper und Musical sind, sind in den Verweisen unten aufgezeigt. Darauf folgen einige Abschnitte, die bestimmte Techniken behandeln, die besonders spezifisch für das Setzen von Singspiel-Partituren sind.

Verweise für Oper und Musical

- Eine große Partitur hat viele gruppierte Systeme und Gesangstext. Arten, Notensysteme zu gruppieren, finden sich in [Systeme gruppieren], Seite 175. Wie Gruppen geschachtelt werden, findet sich in [Verschachtelte Notensysteme], Seite 179.
- Die Ausgabe von leeren Notensystemen in Paritur und Klavierauszug wird oft verhindert. Um eine komprimierte Paritur ohne leere Systeme zu erstellen siehe [Systeme verstecken], Seite 188.
- Wenn die Anzahl der Notensystemgruppen pro Seite sich ändert, wird normalerweise zwischen zwei Systemgruppen ein Trenner gesetzt. Siehe [Systeme trennen], Seite 180.
- Zu Einzelheiten der Seitenformatierung siehe Abschnitt 4.1 [Seitenlayout], Seite 498.
- Stichworte der Dialoge und Regieanweisungen können mit Textbeschriftungen eingefügt werden. Siehe Abschnitt 3.2.3 [Fußnoten erstellen], Seite 465. Ausführliche Regieanweisungen können mit Abschnitten von alleinstehenden Textbeschrifungen zwischen zwei \score-Umgebungen gesetzt werden. Siehe [Separater Text], Seite 222.

Siehe auch

Namens von Figuren

Namens von Figuren werden normalerweise links des Notensystems angezeigt, wenn auf dem System nur die Stimme einer Figure notiert ist:

\score { 
  \new Staff { 
    \set Staff.vocalName = \markup \smallCaps Kaspar 
    \set Staff.shortVocalName = \markup \smallCaps Kas. 
    \relative { 
      \clef "G_8" 
      c'\times4 \times4 \times4 \times4 
      \break 
      c4 \times4 \times4 \times4 
    } 
  } 
} 

\new Staff { 
  \set Staff.vocalName = \markup \smallCaps Melchior 
  \set Staff.shortVocalName = \markup \smallCaps Mel 
  \clef "bass" 
  \relative { 
    a\times4 \times4 \times4 \times4 
    a\times4 \times4 \times4 \times4 
  } 
} 

Wenn zwei oder mehr Figuren sich ein System teilen, wird der Name normalerweise über dem System immer dann gesetzt, wenn der kommende Abschnitt von der Figur gesungen werden soll.
Das kann man mit Textbeschriftungen vornehmen. Oft wird eine bestimmte Schriftart hierfür benutzt.

\relative c' {
  \clef "G_8"
c4 \markup \fontsize #1 \smallCaps Kaspar
  c c c
  \clef "bass"
a4 \markup \fontsize #1 \smallCaps Melchior
  a a a
  \clef "G_8"
c4 \markup \fontsize #1 \smallCaps Kaspar
  c c c
}\relative c' {
  \instrumentSwitch "kaspar"
c4 c c c
  \instrumentSwitch "melchior"
a4 a a a
  \instrumentSwitch "kaspar"
c4 c c c
}

Wenn sehr viele Figurenwechsel vorkommen, kann es auch einfacher sein, „Instrument“-Definitionen für jeden Namen auf oberster Dateiebene zu definieren, sodass \instrumentSwitch der Wechsel der Figur angezeigt werden kann.

\addInstrumentDefinition "kaspar"
\#`((instrumentTransposition . ,(ly:make-pitch -1 0 0))
  (shortInstrumentName . "Kas.")
  (clefGlyph . "clefs.G")
  (clefTransposition . -7)
  (middleCPosition . 1)
  (clefPosition . -2)
  (instrumentCueName . ,(markup #:fontsize 1 #:smallCaps "Kaspar"))
  (midiInstrument . "voice oohs"))

\addInstrumentDefinition "melchior"
\#`((instrumentTransposition . ,(ly:make-pitch 0 0 0))
  (shortInstrumentName . "Mel.")
  (clefGlyph . "clefs.F")
  (clefTransposition . 0)
  (middleCPosition . 6)
  (clefPosition . 2)
  (instrumentCueName . ,(markup #:fontsize 1 #:smallCaps "Melchior"))
  (midiInstrument . "voice aahs"))
Siehe auch

Notationsreferenz: [Instrumentenbezeichnungen], Seite 191, Abschnitt A.21 [Scheme-Funktionen], Seite 759, Abschnitt 1.8 [Text], Seite 215, Abschnitt A.10 [Textbeschriftungsbefehle], Seite 646.

LilyPond Erweitern: Abschnitt "Beschriftungskonstruktionen in Scheme" in Extending.

Musikalische Stichnoten
Stichnoten können in Klavierauszügen, Vocal Books und Orchesterstimmen eingesetzt werden, um anzuziegen, was für Noten eine andere Stimme direkt vor dem eigenen Einsatz spielt. Stichnoten und Instrumentenbezeichnungen werden auch im Klavierauszug eingesetzt, um anzugeben, welches Instrument in der Orchesterfassung den Abschnitt spielt. Das hilft dem Dirigenten, wenn man keine große Partitur parat hat.

Der grundlegende Mechanismus, um Stichnoten einzufügen, findet sich in [Andere Stimmen zitieren], Seite 195, und [Stichnoten formatieren], Seite 198. Aber wenn man viele Stichnoten etwa in einen Klavierauszug einfügen will, um dem Dirigenten zu helfen, muss man sehr sorgfältig mit der Positionierung der Instrumentenbezeichnungen sein. Im folgenden Beispiel gibt es dazu einige Hilfestellungen.

```lilypond
\relative { s4 s4 e'' g } \addQuote "flute" { \flute }
pianoRH = \relative { c''4. g8 } \new CueVoice { \override InstrumentSwitch.self-alignment-X = #RIGHT \set instrumentCueName = "Flute" } \cueDuring "flute" #UP { g4 bes4 }
pianoLH = \relative { c4 <c' e> e, <g c> }
\score { \new PianoStaff << \new Staff { \pianoRH } \new Staff { \clef "bass" \pianoLH } >> }
```


```
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

pianoRH = \relative c'' {
  \transposition c'
  \% position name of cue-ing instrument below the staff
  \new CueVoice {
    \override InstrumentSwitch.self-alignment-X = #RIGHT
    \override InstrumentSwitch.direction = #DOWN
    \set instrumentCueName = "Clar."
  }
  \cueduring "clarinet" #DOWN { c4. g8 }
  g4 bes4
}

pianoLH = \relative { c4 <c' e> e, <g c> }

\score {
  <<
    \new PianoStaff <<
    \new Staff {
      \new Voice {
        \pianoRH
      }
    }
    \new Staff {
      \clef "bass"
      \pianoLH
    }
  >>
  >>
}
Aus diesen zwei Beispielen wird klar, dass es sehr viele Probleme bereiten kann, ausgiebig Stichnoten in einen Klavierauszug einzufügen, und die Noten für das Klavier würden unleserlich. Im folgenden Schnipsel wird jedoch gezeigt, wie man eine Musikfunktion definiert, die Tipparbeit erspart und die Klaviernoten klarer macht.

**Ausgewählte Schnipsel**

*Orchesterstichnoten zu einem Klavierauszug hinzufügen*

Hier wird gezeigt, wie man das Hinzufügen von vielen Orchesterstichnoten zu einem Klavierauszug hinzufügen kann. Die musikalische Funktion `cueWhile` braucht vier Argumente: Die Noten, von denen die Stichnoten formatiert werden sollen, definiert durch `\addQuote`, die Bezeichnung, die mit den Noten angegeben werden soll, dann entweder `#UP` (hoch) oder `#DOWN` (runter) zur Angabe von entweder `\voiceOne` mit der Bezeichnung über dem System oder `\voiceTwo` mit der Bezeichnung unter dem System, und schließlich die Klaviermusik, die parallel zu den Stichnoten gespielt werden soll. Die Bezeichnung des Stichnoteninstruments wird links der Stichnoten positioniert. Viele Abschnitte können zitiert werden, aber sie dürfen sich nicht überlappen.

```latex
cueWhile =
#(define-music-function
  (instrument name dir music)
  (string? string? ly:dir? ly:music?)
#{
  \cueDuring $instrument #dir {
    \once \override TextScript.self-alignment-X = #RIGHT
    \once \override TextScript.direction = $dir
    <>-\markup { \tiny #name }
    $music
  }
} #})

flute = \relative c'' {
  \transposition c'
  s4 s4 e g
} \addQuote "flute" { \flute }

clarinet = \relative c' {
  \transposition bes
  fis4 d d c
} \addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
```
Kapitel 2: Spezielle Notation

\cuelWhile "flute" "Flute" #UP { g4 bes4 } 
\pianoLH = \relative c { c4 <c' e> e, <g c> }
\score {
  \new Staff {
    \new Voice = "singer" {
      \singer
    }
  }
  \new Lyrics {
    \lyricsto "singer"
    \words
  }
  \new PianoStaff <<
    \new Staff {
      \new Voice {
        \pianoRH
      }
    }
    \new Staff {
      \clef "bass"
      \pianoLH
    }
  >>
  >>
}

Siehe auch

Glossar: Abschnitt “cue-notes” in Glossar.

Notationsreferenz: Abschnitt 5.5.1 [Objekte ausrichten], Seite 596, Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581, [Stichnoten formatieren], Seite 198, [Andere Stimmen zitieren], Seite 195, Abschnitt 5.6 [Musikfunktionen benutzen], Seite 605.


**Bekannte Probleme und Warnungen**
\cueduring fügt automatisch einen \texttt{CueVoice}-Kontext ein, in dem alle Noten platziert werden. Das heißt, dass es nicht möglich ist, überlappende Abschnitte an Stichnoten mit dieser Technik zu haben. Überlappende Abschnitte könnten notiert werden, indem man explizit unterschiedliche \texttt{CueVoice}-Kontexte definiert und mit \quote\texttt{During} die Noten ausschneidet und als Stichnoten einfügt.

**Gesprochene Musik**

**Dialog zur Musik**
Dialoge zur Musik wird üblicherweise über den Notensystemen gesetzt, meistens in kursiver Schrift, wobei der Beginn der Phrasen mit einem musikalischen Moment verklammert ist.

```
\relative {
\repeat unfold 3 { a'4 a a a }
}
```

Für längere Abschnitte kann es nötig sein, die Noten zu dehnen, damit die Wörter besser passen. Es gibt keine Möglichkeit, das vollautomatisch von LilyPond erledigen zu lassen, und einige manuelle Änderungen am Seitenlayout sind nötig.


Wenn sich der Dialog über mehr als eine Zeile erstreckt, ist es nötig, manuell Umbrüche mit \texttt{break} einzufügen und die Platzierung des Dialogs anzupassen, damit er nicht in den rechten Seitenrand läuft. Das letzte Wort des letzten Taktes einer Zeile sollte wie oben erklärt getrennt werden.

Hier ein Beispiel, das zeigt, wie das gesetzt werden kann.

```
music = \relative {
\repeat unfold 3 { a'4 a a a }
}

dialogue = \lyricmode {
\markup {
\fontsize #1 \upright \smallCaps Abe: 
"Say this over measures one and"
}4\times7
"two"4 |
\break
"and this over measure"4\times3
```
"three"4 |
}
\score {
  \new Lyrics \with {
    \override LyricText.font-shape = #'italic
    \override LyricText.self-alignment-X = #LEFT
  }
  \{ \dialogue \}
  \new Staff {
    \new Voice { \music }
  }
}

\ABE: *Say this over measures one and two*

\[ \text{and this over measure three} \]

Siehe auch

Notationsreferenz: [Manuelle Silbendauern], Seite 248, Abschnitt 1.8 [Text], Seite 215.
Referenz der Interna: Abschnitt “LyricText” in Referenz der Interna.

2.1.7 Psalmen gesänge und Hymnen

Noten und Text für Psalmen gesänge, Hymnen und Kirchengesänge haben eine spezifische Form in jeder Kirche. Auch wenn die Form sich unterscheidet, sind jedoch die typographischen Probleme sehr ähnlich und werden hier gesammelt behandelt.

Verweise für Psalmen und Hymnen

Wie der Gregorianische Choral in verschiedenen alten Notationsstilen gesetzt wird, findet sich in Abschnitt 2.9 [Notation von alter Musik], Seite 409.

Siehe auch

Notationsreferenz: Abschnitt 2.9 [Notation von alter Musik], Seite 409.

Kirchengesang notieren

Moderne Kirchengesänge benutzen eine Notation mit einer wechselnden Anzahl von Notationselementen der Notation alter Musik. Einige dieser Elemente und Methoden werden hier vorgestellt.

Kirchengesänge werden oft mit Viertelnoten ohne Hälsen notiert, um die Tonhöhen darzustellen, während der Rhythmus sich am Rhythmus der gesprochenen Worte orientiert.

\stemOff = { \hide Staff.Stem }
Kirchengesänge verzichten üblicherweise auf die Taktstriche oder setzen gekürzte oder punktierte Taktstriche ein. Um Taktstriche auszulassen, kann der Bar_engraver entfernt werden.

Taktstriche können auf nur für ein System entfernt werden:
Um Taktstriche nur von einem Abschnitt zu entfernen, kann die Musik als Kadenz notiert werden. Wenn der Abschnitt lang ist, müssen unsichtbare Taktstriche mit \bar "" eingefügt werden, wo Zeilenumbrüche stattfinden sollen.

\relative a' {
  a4 b c2 |
  \cadenza\on
  a4 b c2
  a4 b c2
  \bar ""
  a4 b c2
  a4 b c2
  \cadenza\off
  a4 b c2 |
  a4 b c2 |
}

Pausen können als modifizierte Taktlinien notiert werden:

\relative a' {
  a4
  \cadenza\on
  b c2
  a4 b c2
  \bar """
  a4 b c2
  a4 b c2
  a4 b c2

Alternativ werden die Pausenzeichen der Notation des Gregorianischen Chorals eingesetzt, obwohl die Noten selber modern sind. Das erreicht man durch Veränderung des \breath-Zeichens:

\begin{verbatim}
\score {
relative {
  g'2 a4 g
  \divisioMinima
  g2 a4 g
  \divisioMaior
  g2 a4 g
  \divisioMaxima
  g2 a4 g
  \finalis
}
\layout {
  \context {
    \Staff
    \remove "Bar_engraver"
  }
}
\end{verbatim}
Im Choral wird oft die Taktangabe und teilweise auch der Schlüssel weggelassen.


**SopranoMusic** = \relative {
  g'1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

**AltoMusic** = \relative {
  e'1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

**TenorMusic** = \relative {
  c'1 | c2 c | c1 |
  d1 | g2 g | g g | g1 |
}

**BassMusic** = \relative {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
global = {
\time 2/2
}

% Use markup to center the chant on the page
\markup {
\fill-line {
\score { % centered
<<
\new ChoirStaff <<
\new Staff <<
\global
\clef "treble"
\new Voice = "Soprano" <<
\voiceOne
\SopranoMusic
>>
\new Voice = "Alto" <<
\voiceTwo
\AltoMusic
>>
>>
\new Staff <<
\clef "bass"
\global
\new Voice = "Tenor" <<
\voiceOne
\TenorMusic
>>
\new Voice = "Bass" <<
\voiceTwo
\BassMusic
>>
>>
>>
\layout {
\context {\Score
\override SpacingSpanner.base-shortest-duration = #(ly:make-moment 1/2)
}
\context {\Staff
\remove "Time_signature_engraver"
}
}
}% End score
}% End markup
Einige andere Herangehensweisen derartige Gesänge zu notieren, finden sich im ersten der folgenden Schnipsel.

**Ausgewählte Schnipsel**

*Psalmennotation*

Diese Form der Notation wird benutzt für die Notation von Psalmen, in denen die Strophen nicht die gleiche Länge haben.

\[
\begin{array}{l}
\text{stemOff} = \text{\textbackslash hide Staff.Stem} \\
\text{stemOn} = \text{\textbackslash undo \textbackslash stemOff} \\
\end{array}
\]

\[
\text{\score { \\
\text{\new Staff \with { \remove "Time_signature_engraver" } } \\
\text{\key g \minor} \\
\text{\cadenzaOn} \\
\text{\stemOff a'\breve bes'4 g'4} \\
\text{\stemOn a'2 \bar "||"} \\
\text{\stemOff a'\breve g'4 a'4} \\
\text{\stemOn f'2 \bar "||"} \\
\text{\stemOff a'\breve^\markup { \italic flexe}} \\
\text{\stemOn g'2 \bar "||"} \\
\} \\
}\]

Cantica und andere liturgische Texte können freier gesetzt werden, und können auch Elementer der Notation alter Musik benutzen. Oft werden die Worte unter den Noten und an ihnen ausgerichtet gesetzt. In diesem Fall werden die Noten entsprechend der Ausdehnung des Textes und nicht der Notenlänge gesetzt.

**Vorlage für Alte Notation – moderne Transkription des gregorianischen Chorals**


\[
\text{\include "gregorian.ly"} \\
\text{\begin{array}{l}
\text{chant = \relative c' { \\
\text{\set Score.timing = ##f} \\
\text{f4 a2 \divisioMinima} \\
\text{g4 b a2 f2 \divisioMaior} \\
\text{g4( f) f( g) a2 \finalis} \\
\} } \\
\text{verba = \lyricmode {}} \\
\end{array} \\
}\]

\[
\begin{array}{l}
\text{Cantica and other liturgical texts can be set more freely, and can also use elements of older music notation. Often the words are placed under the notes and aligned with them. In this case, the notes are placed according to the length of the text and not the note length.} \\
\text{A modern transcription of the Gregorian chant. Here there are no bars, no noteheads, and only half and quarter notes are used. Additional symbols indicate the length of rests.} \\
\text{Chant = \relative c' { \\
\text{\set Score.timing = ##f} \\
\text{f4 a2 \divisioMinima} \\
\text{g4 b a2 f2 \divisioMaior} \\
\text{g4( f) f( g) a2 \finalis} \\
\} } \\
\text{Verba = \lyricmode {}} \\
\end{array} \\
\]
Kapitel 2: Spezielle Notation

Siehe auch


Notationsreferenz: Abschnitt 2.9 [Notation von alter Musik], Seite 409, Abschnitt 5.4.6 [Sichtbarkeit von Objekten], Seite 589, Abschnitt 1.2.5 [Takte], Seite 92, [Musik ohne Metrum], Seite 69, Abschnitt 5.1.4 [Umgebungs-Plugins verändern], Seite 558, Abschnitt 2.9.4 [Gregorianischen Choral setzen], Seite 421.

Einen Psalm notieren

Der Text zu einem Anglikanischen Psalm wird normalerweise in separaten Versen zentriert unter den Noten gesetzt.


Einfache Taktstriche (oder in einigen Psalmen ein umgedrehtes Komma) werden zwischen Wörtern eingefügt, um anzuzeigen, wie die Taktstriche der Noten positioniert werden. Im Beschriftungsmodus kann ein einfacher Taktstrich mit | notiert werden.
Kapitel 2: Spezielle Notation

O come let us sing | unto the | Lord : let
us heartily rejoice in the | strength of | our
sal- | -vation.

Andere Symbole benötigen möglicherweise Zeichen aus den fetaMusic-Schriftarten. Siehe Abschnitt 1.8.3 [Schriftarten], Seite 237.

tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}

dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}

tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}

Wenn in einem Takt nur eine ganze Note notiert ist, werden alle Worte dieses Taktes auf dieser Note im Sprechrhythmus gesungen. Wenn im Takt zwei Noten notiert sind, gibt es normalerweise auch nur eine oder zwei Silben. Wenn mehr Silben auf einen Takt gesungen werden sollen, wird mit einem Punkt angegeben, an welcher Stelle die Note gewechselt werden soll.

do
O come let us sing unto the Lord: let us heartily rejoice in the strength of our salvation.

In einigen Psaltern wird ein Asterisk benutzt, um eine Pause in einem rezitierten Abschnitt anzuzeigen, und betonte oder verlängerte Silben werden mit fettem Text angezeigt:

\begin{verbatim}
\column {
  \left-align {
    \line { O come let us sing \tick unto \dot the \tick Lord : let }
    \line { us heartily rejoice in the \tick strength of \tick our }
    \line { sal \tick vation. }
  }
}
\end{verbatim}

Today if ye will hear his voice *
\textbf{harden} | not your | hearts : as in the pro-
\textit{vocation} * and as in the \textbf{day} of tempt-
-\textit{ation} | in the | wilderness.

Andere Psalter setzen einen Akzent über die Silbe, um eine Betonung anzuzeigen:

\begin{verbatim}
tick = \markup {
  \raise #2 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\end{verbatim}
Kapitel 2: Spezielle Notation

Der Einsatz von Beschriftung, um den Text zu zentrieren und die Zeilen in Spalten zu formatieren, findet sich in Abschnitt 1.8.2 [Text formatieren], Seite 223.

Die meisten dieser Elemente werden in einem der beiden Strophen der „Psalm“-Vorlage demonstriert, siehe Abschnitt “Vokalensemble” in Handbuch zum Lernen.

Siehe auch

Notationsreferenz: Abschnitt 1.8.3 [Schriftarten], Seite 237, Abschnitt 1.8.2 [Text formatieren], Seite 223.

Unvollständige Takte in Hymnen

Hymnen beginnen und enden oft jede Zeile der Noten mit einem unvollständigen Takt, sodass jede Notenzeile exakt mit einer Textzeile übereinstimmt. Dazu setzt man den \partial-Befehl zu Beginn der Musik ein und \bar "|" oder \bar "||", um die schließende Taktlinie am Ende der Zeile zu setzen.

\Hymnus-Vorlage

Dieses Beispiel zeigt eine Möglichkeit, eine Hymnusmelodie zu setzen, in der jede Zeile mit einem Auftakt beginnt und einem unvollständigen Takt abschließt. Es zeigt auch, wie man die Strophen als allein stehenden Text unter die Noten hinzufügt.

Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||" \break

O come let us sing | unto the | Lord : let
us heartily rejoice in the | strength of | our
sal- | -vation.

Unvollständige Takte in Hymnen

Hymnen beginnen und enden oft jede Zeile der Noten mit einem unvollständigen Takt, sodass jede Notenzeile exakt mit einer Textzeile übereinstimmt. Dazu setzt man den \partial-Befehl zu Beginn der Musik ein und \bar "|" oder \bar "||", um die schließende Taktlinie am Ende der Zeile zu setzen.

\Hymnus-Vorlage

Dieses Beispiel zeigt eine Möglichkeit, eine Hymnusmelodie zu setzen, in der jede Zeile mit einem Auftakt beginnt und einem unvollständigen Takt abschließt. Es zeigt auch, wie man die Strophen als allein stehenden Text unter die Noten hinzufügt.

Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||" \break
s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||"
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic = \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
  \key g \major
}

\score { % Start score
  \global
  \new PianoStaff << % Start pianostaff
  \new Staff << % Start Staff = RH
    \clef "treble"
    \new Voice = "Soprano" << % Start Voice = "Soprano"
      \Timeline
      \voiceOne
      \SopranoMusic
    >> % End Voice = "Soprano"

    \new Voice = "Alto" << % Start Voice = "Alto"
      \Timeline
      \voiceTwo
      \AltoMusic
    >> % End Voice = "Alto"

  >> % End Staff = RH

  \new Staff << % Start Staff = LH
    \global
    \clef "bass"
    \new Voice = "Tenor" << % Start Voice = "Tenor"
      \Timeline
      \voiceOne
      \TenorMusic
    >> % End Voice = "Tenor"
\new Voice = "Bass" << % Start Voice = "Bass"
\Timeline
\voiceTwo
\BassMusic
\BassMusic
>> % End Voice = "Bass"
>> % End Staff = LH
>> % End pianostaff
}
% End score

\markup {
\fill-line {
"
{
\column {
\left-align {
"This is line one of the first verse"
"This is line two of the same"
"And here's line three of the first verse"
"And the last line of the same"
}
}
"
}
}

\paper { % Start paper block
indent = 0 % don't indent first system
line-width = 130 % shorten line length to suit music
}
% End paper block

This is line one of the first verse
This is line two of the same
And here's line three of the first verse
And the last line of the same
2.1.8 Alte Vokalmusik

Alte Vokalmusik ist unterstützt, wie erklärt in Abschnitt 2.9 [Notation von alter Musik], Seite 409.

Siehe auch

Notationsreferenz: Abschnitt 2.9 [Notation von alter Musik], Seite 409.

2.2 Tasteninstrumente und andere Instrumente mit mehreren Systemen

Dieser Abschnitt behandelt verschiedene Notationsaspekte, die typischerweise in Noten für Tasteninstrumente und andere Instrumente auf mehreren Notensystemen auftreten, wie etwa
Harfe und Vibraphon. Hier wird die gesamte Gruppe von Instrumenten, die auf mehreren Systemen notiert werden, als „Tasteninstrumente“ bezeichnet, auch wenn einige von ihnen keine Tasten aufweisen.

2.2.1 Übliche Notation für Tasteninstrumente

Dieser Abschnitt zeigt allgemeine Eigenschaften des Notensatzes, die für die meisten Instrumente mit mehreren Systemen benötigt werden.

Referenz für Tasteninstrumente

Tasteninstrumente werden normalerweise auf einem Klaviersonsystem notiert. Es besteht aus zwei Notensystemen, die durch eine Klammer verbunden sind. Die gleiche Notation wird auch für andere Tasteninstrumente sowie Harfen verwendet. Orgelmusik wird normalerweise auf zwei Systemen innerhalb eines Klaviersonsystems notiert, denen noch ein drittes normales Notensystem für die Pedaltöne hinzugefügt wird.

Die Systeme eines Klaviersonsystems sind ziemlich unabhängig, aber Stimmen können bei Bedarf zwischen den Systemen wechseln.

Einige häufige Besonderheiten von Notation für Tasteninstrumenten wird an anderen Stellen besprochen:

- Noten für Tasteninstrumente haben oft mehrere Stimmen und die Anzahl der Stimmen kann sich häufig ändern. Das ist beschrieben in [Auflösung von Zusammenstößen], Seite 161.
- Noten für Tasteninstrumente kann auch parallel, Takt für Takt notiert werden, wie gezeigt in [Musik parallel notieren], Seite 171.
- Dynamikbezeichnung kann in einem Dynamics-Context notiert werden, der zwischen zwei Staff-Kontexten steht und dann horizontal zwischen diesen beiden zentriert wird; siehe [Dynamik], Seite 116.
- Fingersatz wird erklärt in [Fingersatzanweisungen], Seite 205.
- Orgelpedal-Zeichen werden als Artikulationszeichen notiert, siehe Abschnitt A.13 [Liste der Artikulationszeichen], Seite 705.
- Vertikale Rasterlinien können erstellt werden, siehe [Gitternetzlinien], Seite 212.
- Noten für Tasteninstrumente beinhalten oft Laissez vibre-Bögen und Bindebögen mit Arpeggio oder Tremolo, siehe hierzu [Bindebögen], Seite 49.
- Arpeggios können auch zwischen den Systemen verbunden werden, siehe hierzu [Arpeggio], Seite 130.
- Tremolo-Zeichen finden sich in [Tremolo-Wiederholung], Seite 149.
- Viele der Optimierungen, die für Tastenmusik nötig sein können, sind demonstriert in Abschnitt “Beispiel aus dem Leben” in Handbuch zum Lernen.
- Unsichtbare Noten können eingesetzt werden, um Überbindungen zwischen Stimmen zu setzen, siehe Abschnitt “Andere Benutzung von Optimierungen” in Handbuch zum Lernen.

Siehe auch


Notationsreferenz: [Systeme gruppieren], Seite 175, [Instrumentenbezeichnungen], Seite 191, [Auflösung von Zusammenstößen], Seite 161, [Musik parallel notieren], Seite 171, [Fingersatzanweisungen], Seite 205, Abschnitt A.13 [Liste der Artikulationszeichen], Seite 705, [Gitternetzlinien], Seite 212, [Bindebögen], Seite 49, [Arpeggio], Seite 130, [Tremolo-Wiederholung], Seite 149.


Referenz der Interna: Abschnitt “PianoStaff” in Referenz der Interna.
Notensysteme manuell verändern

Stimmen können mit dem Befehl

\change Staff = Systembezeichnung

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", "RH" oder "LH".

Das System, zu dem die Stimme wechseln soll, muss zum Zeitpunkt des Wechsels existieren. Wenn notwendig, müssen Systeme "künstlich am Leben gehalten werden", siehe Abschnitt 5.1.3 [Kontexte am Leben halten], Seite 555.

Balken zwischen den Systemen werden automatisch erstellt:

```
\new PianoStaff <<
\new Staff = "up" {
  e' c'>8
  \change Staff = "down"
  g8 fis g
  \change Staff = "up"
  g'' c''>8
  \change Staff = "down"
  e8 dis e
  \change Staff = "up"
}
\new Staff = "down" {
  \clef bass
  % keep staff alive
  s1
}
>>
```

Wenn die Balken verändert werden müssen, sollte zuerst die Richtung des Balkens beeinflusst werden. Die Balkenposition wird dann von der Mitte des Systems gemessen, dass näher am Balken ist. Ein einfaches Beispiel ist gezeigt in Abschnitt "Überlappende Notation in Ordnung bringen" in *Handbuch zum Lernen*.

Bei Stimmen, die zwischen den Systemen wechseln, kann es zu überlappenden Notation kommen:

```
\new PianoStaff <<
\new Staff = "up" {
  \voiceOne
  % Make space for fingering in the cross-staff voice
  \once\override DynamicLineSpanner.staff-padding = #3.4
  e''\p\< d''\>
  c''!\}
\new Staff = "down" <<
```
Die Hälsen und Bögen überlappen sich mit der dazwischenstehenden Dynamik-Zeile, weil die automatische Zusammenstoßauflösung für Balken, Bögen und andere Strecker, die Noten zwischen unterschiedlichen Systemen verbinden, ausgeschaltet ist. Das gilt auch für Hälsen und Artikulationszeichen, wenn ihre Positionierung durch einen Strecker zwischen Systemen verändert wird. Die resultierenden Zusammenstöße müssen manuell aufgelöst werden, wo es nötig ist, dabei kann man die Methoden anwenden, die in Abschnitt “Überlappende Notation in Ordnung bringen” in Handbuch zum Lernen gezeigt werden.

Siehe auch

Handbuch zum Lernen: Abschnitt “Überlappende Notation in Ordnung bringen” in Handbuch zum Lernen.

Notationsreferenz: [Hälsen], Seite 210, [Automatische Balken], Seite 77, Abschnitt 5.1.3 [Kon- texte am Leben halten], Seite 555.

Bekannte Probleme und Warnungen

Die Zusammenstoßauflösung für Balken funktioniert nicht für Balken, die direkt am Ende eines Systems enden. In diesem Fall muss man manuelle Balken einsetzen.

Automatischer Systemwechsel

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

\autochange ...Noten...

Ein \relative-Abschnitt, der sich außerhalb des \autochange-Abschnittes befindet, hat keinen Einfluss auf die Notenhöhen.

Wenn individuelle Kontrolle über die einzelnen Systeme benötigt wird, können sie manuell mit den Bezeichnungen "up" und "down" erstellt werden. Der \autochange-Befehl wechselt dann die Stimme zwischen den Systemen.

\begin{Verbatim}
Achtung: Wenn Systeme manuell erstellt werden, \textbf{müssen} sie genau die Bezeichnungen "up" und "down" bekommen, damit die automatische Wechselfunktion sie erkennen kann.
\end{Verbatim}

Systeme müssen etwa manuell erstellt werden, damit die Tonart im unteren System gesetzt werden kann:

\begin{Verbatim}
\new PianoStaff { 
\new Staff = "up" { 
\new Voice = "melodieEins" { 
\key g \major 
\autochange \relative { 
g8 b a c b d c e 
d8 r fis, g a2 
} 
} 
\new Staff = "down" { 
\key g \major 
\clef bass 
} 
} >>
\end{Verbatim}
Siehe auch
Notationsreferenz: [Notensysteme manuell verändern], Seite 305.

Bekannte Probleme und Warnungen
Die Aufteilung auf die Systeme geschieht nicht unbedingt an optimaler Stelle. Für bessere Qualität müssen die Wechsel manuell eingestellt werden.

Akkorde werden nicht über die Systeme verteilt, sie werden dem System zugewiesen, auf dem sich ihre erste Note befinden würde.

Stimmführlingslinien
Immer, wenn eine Stimme von einem Klaviersystem zu dem anderen wechselt, kann automatisch eine Linie zur Verdeutlichung des Stimmenverlaufs ausgegeben werden:

\new PianoStaff <<
\new Staff = "one" {
 \showStaffSwitch
  c'1
 \change Staff = "two"
  b2 a
}
\new Staff = "two" {
 \clef bass
  s1*2
}
>>

Vordefinierte Befehle
\showStaffSwitch, \hideStaffSwitch.

Siehe auch
Referenz der Interna: Abschnitt “Note_head_line_engraver” in Referenz der Interna,
Abschnitt “VoiceFollower” in Referenz der Interna.

Hälse über beide Systeme
Akkorde, die über zwei Systeme reichen, können mit dem Span_stem_engraver erstellt werden. Man muss dabei sicherstellen, dass die automatische Bebalkung die Noten nicht auf dem einen System mit Balken versieht, wenn es auf dem anderen nicht nötig wäre.

\layout {
  \context {
    \PianoStaff
Ausgewählte Schnipsel

Akkorde auf zwei Systemen mit Arpeggioklammern anzeigen

Eine Arpeggioklammer kann anzeigen, dass Noten auf zwei unterschiedlichen Systemen mit der selben Hand gespielt werden sollen. Damit das notiert werden kann, muss der PianoStaff-Kontext so eingestellt werden, dass er Arpeggios über Systeme hinweg akzeptiert und die Form der Arpeggios muss auf eine Klammer eingestellt werden.

(Debussy, Les collines d’Anacapri, T. 65)
Kapitel 2: Spezielle Notation

2.2.2 Klavier

Dieser Abschnitt zeigt Eigenheiten der Notation von Klavermusik

Klavierpedal

Klaviere (teilweise auch Vibraphone und Celesta) besitzen üblicherweise drei Pedale, das linke oder Haltepedal, das rechte oder Una-corda-Pedal und das Sostenuto-Pedal. Die englischen Be- griff hierzu lauten: sustain, sostenuto und una corda.

```
\relative {
    c'4\sustainOn d e g
    <c, f a>1\sustainOff
    c4\sostenutoOn e g c,
    <bes d f>1\sostenutoOff
    c4\unaCorda d e g
    <d fis a>1\treCorde
}
```

Die Pedalbezeichnung kann auf drei Arten vorgenommen werden: mit Text, Klammern oder einer Mischung aus beidem. Das Haltepedal und das Una-corda-Pedal benutzen als Standard die Textdarstellung, während das Sostenuto-Pedal den gemischten Stil benutzt:

```
\relative {
```
Die Platzierung der Befehle entspricht der Bewegung der Pedale während des Spiels. Um das Pedal bis zur letzten Taktkante zu halten, muss der letzte Pedal-hoch-Befehl weggelassen werden.

Pedalbezeichnungen können innerhalb eines Dynamics-Kontextes notiert werden, sodass sie an einer horizontalen Linie ausgerichtet werden.

Siehe auch
Notationsreferenz: [Bindebogen], Seite 49.

2.2.3 Akkordeon
Dieser Abschnitt behandelt Notation, die nur für Akkordeonmusik benötigt wird.

Diskant-Symbole
Akkordeons werden oft mit mehreren Reihen an Zungen gebaut, welche Unisono oder eine Oktave höher bzw. tiefer erklingen. Jedes Akkordeon hat eigene Bezeichnungen für die Register (engl. shift) wie etwa Oboe, Bandonium usw. Eine Anzahl an Symbolen wird benutzt um die Wechsel anzuzeigen.

Ausgewählte Schnipsel
*Symbole für Akkordeon-Diskantregister*
Diskantregister für Akkordeon können mit \markup dargestellt werden. Die vertikale Position der einzelnen Elemente werden mit \raise angepasst.

\[\text{\texttt{discant = \markup \{ \newline \musicglyph "accordion.discant" \}}}\]
\[\text{\texttt{dot = \markup \{ \newline \musicglyph "accordion.dot" \}}}\]
\layout { ragged-right = ##t }

% 16 voets register
accBasson = \markup {
    \combine
    \discant
    \translate-scaled #'(0 . 0.5) \dot
}

% een korig 8 en 16 voets register
accBandon = \markup {
    \combine
    \discant
    \combine
    \translate-scaled #'(0 . 0.5) \dot
    \translate-scaled #'(0 . 1.5) \dot
}

accVCello = \markup {
    \combine
    \discant
    \combine
    \translate-scaled #'(0 . 0.5) \dot
    \combine
    \translate-scaled #'(0 . 1.5) \dot
    \translate-scaled #'(1 . 1.5) \dot
}

% 4-8-16 voets register
accHarmon = \markup {
    \combine
    \discant
    \combine
    \translate-scaled #'(0 . 0.5) \dot
    \combine
    \translate-scaled #'(0 . 1.5) \dot
    \translate-scaled #'(0 . 2.5) \dot
}

accTrombon = \markup {
    \combine
    \discant
    \combine
    \translate-scaled #'(0 . 0.5) \dot
    \combine
    \translate-scaled #'(0 . 1.5) \dot
    \combine
    \translate-scaled #'(1 . 1.5) \dot
    \translate-scaled #'(-1 . 1.5) \dot
}
% eenkorig 4 en 16 voets register
accOrgan = \markup {
  \combine
  \discant
  \combine
    \translate-scaled #'(0 . 0.5) \dot
    \translate-scaled #'(0 . 2.5) \dot
}

accMaster = \markup {
  \combine
  \discant
  \combine
    \translate-scaled #'(0 . 0.5) \dot
    \translate-scaled #'(0 . 1.5) \dot
    \translate-scaled #'(1 . 1.5) \dot
    \translate-scaled #'(0 . 2.5) \dot
}

accAccord = \markup {
  \combine
  \discant
  \combine
    \translate-scaled #'(0 . 1.5) \dot
    \translate-scaled #'(1 . 1.5) \dot
    \translate-scaled #'(0 . 2.5) \dot
}

accMusette = \markup {
  \combine
  \discant
  \combine
    \translate-scaled #'(0 . 1.5) \dot
    \translate-scaled #'(0 . 2.5) \dot
}

accCeleste = \markup {
  \combine
  \discant
  \combine
    \translate-scaled #'(0 . 1.5) \dot
    \translate-scaled #'(0 . 2.5) \dot
accOboe = ^\markup {
  \combine
  \discant
  \combine
    \translate-scaled #'(0 . 1.5) \dot
    \translate-scaled #'(0 . 2.5) \dot
}

accClarin = ^\markup {
  \combine
  \discant
    \translate-scaled #'(0 . 1.5) \dot
}

accPiccolo = ^\markup {
  \combine
    \discant
      \translate-scaled #'(0 . 2.5) \dot
}

accViolin = ^\markup {
  \combine
    \discant
      \translate-scaled #'(0 . 1.5) \dot
      \combine
        \translate-scaled #'(1 . 1.5) \dot
        \translate-scaled #'(0 . 2.5) \dot
}

\relative c'' {
  c4 d\accBasson e f
c4 d\accBandon e f
c4 d\accVCello e f
c4 d\accHarmon e f
c4 d\accTrombon e f
  \break
  c4 d\accOrgan e f
c4 d\accMaster e f
c4 d\accAccord e f
c4 d\accMusette e f
c4 d\accCeleste e f
  \break
  c4 d\accOboe e f
c4 d\accClarin e f
c4 d\accPiccolo e f
c4 d\accViolin e f
}
2.2.4 Harfe

Dieser Abschnitt zeigt Eigenheiten der Notation für Harfe.

Referenzen für Harfe

Einige übliche Notationseigenheiten für Harfe sind woanders behandelt:
- Glissando ist die üblichste Harfentechnik, siehe [Glissando], Seite 129.
- Ein Bisbigliando wird als ein Tremolo notiert, siehe [Tremolo-Wiederholung], Seite 149.
- Flageolettöne werden hier beschrieben: [Flageolett], Seite 318.
- Für Arpeggio und non-arpeggio, siehe [Arpeggio], Seite 130.

Siehe auch

Notationsreferenz: [Tremolo-Wiederholung], Seite 149, [Glissando], Seite 129, [Arpeggio], Seite 130, [Flageolett], Seite 318.

Harfenpedal

Harfe haben sieben Saiten in einer Oktave, die entweder als normaler Ton, oder aber erhöht bzw. erniedrigt klingen können. Bei einer Hakenharfe kann man jede Saite einzeln einstellen, bei Pedalharfen aber wird jede Saite mit der gleichen Notenbezeichnung von einem einzigen Pedal kontrolliert. Vom Spieler aus gesehen von rechts nach links sind die Pedale: D, C und H für die linke und E, F, G und A für die rechte Seite. Die Position des Pedals kann mit Textbeschriftungselementen:
\textLengthOn
cis'"1\markup \concat \vcenter {  
  [D \flat C \sharp B|E \sharp F \sharp G A \flat] } 
c'"!1\markup \concat \vcenter {  
  [ C \natural ] } 
\textLengthOn

oder Pedaldiagrammen angezeigt werden:
\textLengthOn

Kapitel 2: Spezielle Notation

Der \harp-pedal-Befehl braucht eine Anzahl an Zeichen, von welchen \^ die höchste Pedalposition (erniedrigte Tonhöhe), - die mittlere Pedalposition (normale Tonhöhe, v die tiefste Pedalposition (erhöhter Ton) anzeigt. | ist ein Trenner. Ein o vor der Definition umrandet das Symbol.

Siehe auch

Notationsreferenz: [Textarten], Seite 216, Abschnitt A.10.5 [Instrument Specific Markup], Seite 686.

2.3 Bundlose Saiteninstrumente

Dieser Abschnitt stellt Information und Referenzen zur Verfügung, die beim Setzen von Noten für Saiteninstrumente ohne Bund herangezogen werden können.

2.3.1 Übliche Notation für bundlose Saiteninstrumente

Hinweise für bundlose Saiteninstrumente

Die meisten Notationseigenschaften, die für Orchesterstreicher eingesetzt werden, sind an anderer Stelle beschrieben:

- Textanweisungen wie „pizz.“ oder „arco“ werden als einfacher Text eingefügt, siehe [Textarten], Seite 216.
- Fingersatz, auch das Zeichen für den Daumen, ist erklärt in [Fingersatzanweisungen], Seite 205.
- Doppelgriffe werden normalerweise als Akkord notiert, siehe hierzu [Noten mit Akkorden], Seite 152. Anweisungen, wie Akkorde gespielt werden sollen, können auch hinzugefügt werden, siehe [Arpeggio], Seite 130.
- Eine Vorlage für Streichquartett findet sich in Abschnitt “Streichquartett” in Handbuch zum Lernen. Andere sind als Schnipsel zur Verfügung gestellt.

Siehe auch

Handbuch zum Lernen: Abschnitt “Streichquartett” in Handbuch zum Lernen.

Notationsreferenz: [Textarten], Seite 216, [Fingersatzanweisungen], Seite 205, [Noten mit Akkorden], Seite 152, [Arpeggio], Seite 130.


Bezeichnung des Bogens

Hinweise zur Bogenführung können als Artikulationen erstellt werden, wie beschrieben in [Artikulationszeichen und Verzierungen], Seite 113.

Die Befehle \upbow und \downbow werden mit Legatobögen in folgender Weise eingesetzt:

\relative { c'4(\downbow d) e(\upbow f) }

und das nächste Beispiel zeigt drei Arten, eine offene A-Saite auf der Geige anzuzeigen:

a4 \open
a^\markup { \teeny "II" }
a2^\markup { \small "sul A" }

Vordefinierte Befehle

\downbow, \upbow, \open.

Siehe auch

Notation Reference: [Artikulationszeichen und Verzierungen], Seite 113, [Legatobögen], Seite 123.
Flageolett

Natürliches Flageolett

Flageolett-Töne können auf verschiedene Arten notiert werden. Üblicherweise werden sie mit einem Rautenkopf notiert, wenn ein Ton angezeigt werde, bei dem die Saiten berührt wird, wo sie sonst abgegriffen würde.

```
\relative d'' {
  d4 e4.
  \harmonicsOn
  d8 e e
  d4 e4.
  \harmonicsOff
  d8 e e
}
```

Alternativ kann auch eine normale Noten die Tonhöhe anzeigen, die erklingen soll, wobei ein kleiner Kreis angibt, dass es sich um einen Flageolett-Ton handelt:

```
d''2 ^\flageolet d'' _\flageolet
```

Künstliches Flageolett

Künstliche Flageolettöne werden mit zwei Noten notiert, von denen einen einen normalen Notenkopf besitzt und die Griffposition des Fingers angibt, während die andere in Rautenform die Position des leicht aufgesetzten Fingers anzeigt.

```
\relative e' {
  <e a \harmonic>2. <c g \harmonic>4
  \set harmonicDots = ##t
  <e a \harmonic>2. <c g \harmonic>4
}
```

Achtung: `\harmonic` muss innerhalb einer Akkordkonstruktion gesetzt werden, auch wenn nur eine Note gesetzt wird. Normalerweise würde `\harmonicsOn` in dieser Situation benutzt.

Siehe auch

Glossar: Abschnitt ”harmonics” in Glossar.

Notationsreferenz: [Besondere Notenköpfe], Seite 34, [Hinweise für bundlose Saiteninstrumente], Seite 317.
Bartók-Pizzicato

Ein Knallpizzicato, auch als Bartók-Pizzicato bekannt, ist ein hartes Pizzicato, bei dem man die Saite nach oben (und nicht seitlich) zieht, sodass sie beim Schwingen das Griffbrett berührt.

\relative {
c'4\snappizzicato
<c' e g>4\snappizzicato
<c' e g>4\snappizzicato
<c, e g>4\snappizzicato
}

2.4 Saiteninstrumente mit Bünden
Dieser Abschnitt erklärt bestimmte Eigenheiten der Notation für Saiteninstrumente mit Bünden.

2.4.1 Übliche Notation für Saiteninstrumente mit Bünden

Dieser Abschnitt zeigt Besonderheiten der Notation, die allen Bündinstrumenten eigen ist.

Referenz für Saiteninstrumente mit Bünden

Noten für Bündinstrumente wird normalerweise auf einem einzelnen System notiert, entweder als traditionelles Notensystem oder in Tabulaturform. Manchmal werden beide Arten miteinander verbunden, und besonders in populärer Musik ist es üblich, über dem traditionellen System Griffsymbole zu setzen. Gitarre und Banjo sind transponierende Instrumente, die eine Oktave tiefer klingen als sie notiert werden. Partituren für diese Instrumente sollten den „Tenorschlussel“ ("treble_8" bzw. \transposition c) benutzen, um korrekte MIDI-Dateien zu erhalten. Einige Spezifika für Instrumente mit Bünden sind an anderer Stelle erklärt:

- Fingersatz kann notiert werden, siehe [Fingersatzanweisungen], Seite 205.
- Anweisungen für Laissez vibrer-Bögen und Bögen zwischen Arpeggios und Tremolos sind beschrieben in [Bindebögen], Seite 49.
- Hinweise, wie mehrere Stimmen gesetzt werden können, finden sich in [Auflösung von Zusammenstößen], Seite 161.
- Instructions for indicating harmonics can be found in [Flageolett], Seite 318.

Siehe auch

Notationsreferenz: [Fingersatzanweisungen], Seite 205, [Bindebögen], Seite 49, [Auflösung von Zusammenstößen], Seite 161, [Instrumentenbezeichnungen], Seite 191, [Musik parallel notieren], Seite 171, [Arpeggio], Seite 130, Abschnitt A.13 [Liste der Artikulationszeichen], Seite 705, [Notenschlüssel], Seite 16, [Transposition von Instrumenten], Seite 23.

Seitennummerbezeichnung

Die Nummer der Saitе, auf der gespielt werden soll, kann angezeigt werden, indem \Zahl an eine Note gehängt wird:

\clef "treble_8"
\c4\5 e\4 g2\3
<c\5 e\4 g\3>1

\clef "treble_8"
\c4\5 e\4 g2\3
<c\5 e\4 g\3>1

```
\clef "treble_8"
g4\3-0
g-0\3
<g\3-0>
<g-0\3>
```

![Musiknotenschlüssel](image)

Ausgewählte Schnipsel

*Position von Fingersatz in Akkorden kontrollieren*

Die Position von Fingersatzzahlen kann exakt kontrolliert werden.

```
\relative c' {  
  \set fingeringOrientations = #'(left)  
  <c-1 e-3 a-5>4  
  \set fingeringOrientations = #'(down)  
  <c-1 e-3 a-5>4  
  \set fingeringOrientations = #'(down right up)  
  <c-1 e-3 a-5>4  
  \set fingeringOrientations = #'(up)  
  <c-1 e-3 a-5>4  
  \set fingeringOrientations = #'(left)  
  <c-1>2  
  \set fingeringOrientations = #'(down)  
  <e-3>2  
}
```

![Musiknotenschlüssel](image)

*Fingersatz auch innerhalb des Systems setzen*

Normalerweise werden vertikal orientierte Fingersatzzahlen außerhalb des Systems gesetzt. Das kann aber verändert werden.

```
\relative c' {  
  <c-1 e-2 g-3 b-5>2  
  \override Fingering.staff-padding = #'()  
  <c-1 e-2 g-3 b-5>4 <g'-0>  
}
```
Siehe auch

Notationsreferenz: [Fingersatzanweisungen], Seite 205.


Referenz der Interna: Abschnitt “StringNumber” in Referenz der Interna, Abschnitt “Fingering” in Referenz der Interna.

Standardtabulaturen

Musik für gezupfte Saiteninstrumente wird oft notiert, indem man eine Finger/Berührungsnotation bzw. Tabulatur benutzt. Im Gegensatz zur traditionellen Notation werden hier Tonhöhen nicht mit Notenköpfen notiert, sondern mit Zahlen (oder buchstabenzählichen Symbolen in historischen Tabulaturen). Die Notenlinien einer Tabulatur zeigen die Saiten an, auf der eine Note gespielt werden soll, und eine Zahl auf einer Notenlinie zeigt an, welcher Bund für eine Note gespielt werden muss. Die Zahlen werden vertikal übereinander geschrieben, wenn sie gleichzeitig gespielt werden sollen.

Standardmäßig ist Saiten 1 die höchste Saiten und entspricht der höchsten Notenlinie des TabStaff (der Tabulatur). Die voreingestellte Saitenstimme der Tabulatur ist die normale Gitarrenstimme (mit 6 Saiten). Die Noten werden als Tabulatur ausgegeben, wenn man den TabStaff-Kontext und darin den TabVoice-Kontext benutzt. Ein kalligraphischer Tabulaturschlüssel wird automatisch hinzugefügt.

\new TabStaff \relative {
  a,8 a' <c e> a
  d,8 a' <d f> a
}

Standard-Tabulaturen haben weder Symbole, die Notendauern anzeigen, noch andere musikalische Symbole wie etwa Ausdrucksbezeichnungen.

symbols = {
  \time 3/4
  c4-."Allegro" d( e)
  f4-\.f g a\"fermata
  \mark \default
  c8_..\}\c16 c" 2!\c'2.\prell1\}
}

\score {
  <<
    \new Staff \clef "G_8" \symbols }
    \new TabStaff \symbols }
  >>
}
Wenn alle musikalischen Symbole, die in der traditionellen Notation eingesetzt werden, auch in der Tabulatur gedruckt werden sollen, muss man den Befehl \tabFullNotation in einem TabStaff-Kontext hinzufügen. Dabei ist zu beachten, dass halbe Noten in einer Tabulatur mit zwei Hälsen dargestellt werden, um sie von Viertelnoten zu unterscheiden.

```
symbols = {
    \time 3/4
    c4-."Allegro" d\( e \)
    f4- f g a\( \text{fermata} \)
    \mark \default
    c8_.\( c16 c^-2! \)
    c'2.\prall\}
}
```

```
\score {
\new TabStaff {
    \tabFullNotation
    \symbols
}
}
```

Normalerweise werden Tonhöhen der tiefstmöglichen Spielposition auf dem Bundbrett zugewiesen (erste Lage). Offene Saiten werden automatisch bevorzugt. Wenn man eine bestimmte Tonhöhe auf einer bestimmten Saiten gespielt haben will, kann man eine Saitennummeranweisung zur Tonhöhe hinzufügen. Wenn man die Saitenzahlanweisung nicht in der traditionellen Notation sehen will, kann man den entsprechenden Stencil mit \override verändern. Es ist jedoch sehr viel bequemer, die Spielposition unter Benutzung von minimumFret zu definieren. Der Standardwert von minimumFret beträgt 0.

Auch wenn minimumFret gesetzt ist, werden offene Saiten immer benützt, wenn es möglich ist. Dieses Verhalten kann verändert werden, indem restrainOpenStrings auf #t gesetzt wird.

```
\layout { \omit Voice.StringNumber }
\new StaffGroup <<
    \new Staff \relative {
        \clef "treble_8"
        \time 2/4
        c16 d e f g4
c,16\( d\( e\( f\)4 \)g4\)
c,16 d e f g4
    }
```
Akkord-Konstruktionen können mit dem Akkord-Wiederholungssymbol \texttt{\textbackslash chordRepeats '#\textbackslash (string-number-event fingering-event)'} explizit für musikalische Ausdrücke in Tabulaturen aufrufen, wenn Akkordwiederholungen gewünscht sind. Der Befehl ist so wichtig, dass er durch \texttt{\textbackslash tabChordRepeats} zur Verfügung gestellt wird.

\texttt{\textbackslash chordRepeats '#'(string-number-event fingering-event)}

...
Der Befehl \hideSplitTiedTabNotes hebt das Verhalten auf, dass Bundnummern in Klammern gesetzt werden:

ties = \relative {
    \repeat volta 2 {
        e'2. f4~
        2 g2~
    }
}
Flageolett (engl. harmonic) kann zur Tabulaturnotation als klingende Tonhöhe hinzugefügt werden:

```
\layout { \omit Voice.StringNumber }
firstHarmonic = {
  d'4\4\harmonic
  g'4\3\harmonic
```


\fretHarmonics = {
  \harmonicByFret #5 d16\4
  \harmonicByFret #4 d16\4
  \harmonicByFret #3 d8\4
  \harmonicByFret #5 <g\3 b\2>2.
}

Alternativ können Flageolett-Töne auch errechnet werden, indem man das Verhältnis der Saitenlängen über und unter dem Flageolett-Finger definiert:
Ausgewählte Schnipsel

_Hals- und Balkenverhalten in einer Tabulatur_


\new TabStaff {
  \relative c {
    \tabFullNotation
    g16 b d g b d g b
    \stemDown
    \override Beam.concaveness = #10000
    g,,,16 b d g b d g b
  }
}

_Polyphonie in einer Tabulatur_

Polyphonie kann in einer Tabulatur (TabStaff) genauso wie in einem normalen Notensystem erstellt werden.

upper = \relative c' {
  \time 12/8
Referenz für Flageolett von offenen Saiten

Referenz für Flageolett von offenen Saiten:

```latex
openStringHarmonics = {
    \textSpannerDown
    \override TextSpanner.staff-padding = #3
    \override TextSpanner.dash-fraction = #0.3
    \override TextSpanner.dash-period = #1

    %first harmonic
    \override TextSpanner.bound-details.left.text = \markup\small "1st harm."
    \harmonicByFret #12 e,2\6\startTextSpan
    \harmonicByRatio #1/2 e,\6\stopTextSpan

    %second harmonic
    \override TextSpanner.bound-details.left.text = \markup\small "2nd harm."
```
Kapitel 2: Spezielle Notation

\harmonicByFret #7 e,\6\startTextSpan
\harmonicByRatio #1/3 e,\6
\harmonicByFret #19 e,\6
\harmonicByRatio #2/3 e,\6\stopTextSpan
\%\harmonicByFret #19 < e,\6 a,\5 d\4 >
\%\harmonicByRatio #2/3 < e,\6 a,\5 d\4 >

\%third harmonic
\override TextSpanner.bound-details.left.text = \markup\small "3rd harm. "
\harmonicByFret #5 e,\6\startTextSpan
\harmonicByRatio #1/4 e,\6
\harmonicByFret #24 e,\6
\harmonicByRatio #3/4 e,\6\stopTextSpan
\break

\%fourth harmonic
\override TextSpanner.bound-details.left.text = \markup\small "4th harm. "
\harmonicByFret #4 e,\6\startTextSpan
\harmonicByRatio #1/5 e,\6
\harmonicByFret #9 e,\6
\harmonicByRatio #2/5 e,\6
\harmonicByFret #16 e,\6
\harmonicByRatio #3/5 e,\6\stopTextSpan

\%fifth harmonic
\override TextSpanner.bound-details.left.text = \markup\small "5th harm. "
\harmonicByFret #3 e,\6\startTextSpan
\harmonicByRatio #1/6 e,\6\stopTextSpan
\break

\%sixth harmonic
\override TextSpanner.bound-details.left.text = \markup\small "6th harm. "
\harmonicByFret #2.7 e,\6\startTextSpan
\harmonicByRatio #1/7 e,\6\stopTextSpan

\%seventh harmonic
\override TextSpanner.bound-details.left.text = \markup\small "7th harm. "
\harmonicByFret #2.3 e,\6\startTextSpan
\harmonicByRatio #1/8 e,\6\stopTextSpan

\%eighth harmonic
\override TextSpanner.bound-details.left.text = \markup\small "8th harm. "
\harmonicByFret #2 e,\6\startTextSpan
\harmonicByRatio #1/9 e,\6\stopTextSpan
}

\score { <<
   \new Staff
   \with { \omit StringNumber } { 
      \new Voice { 
         \clef "treble_8" 
   } 
}
Flageolett von Bundinstrumenten in einer Tabulatur

Flageolett für Bundinstrumente:

```
pinchedHarmonics = {
  \textSpannerDown
  \override TextSpanner.bound-details.left.text =
  \markup {\halign #-.5 \teeny "PH" }
  \override TextSpanner.style =
  #'dashed-line
  \override TextSpanner.dash-period = #0.6
  \override TextSpanner.dash-period = #0.6
  \override TextSpanner.dash-period = #0.6
  \markup {\draw-line #'(0 . 1)}
  \override TextSpanner.bound-details.right.padding = #-0.5
}
```
harmonics = {
  %artificial harmonics (AH)
  \textLengthOn
  <<<\parenthesize b b'\harmonic>4\markup { \teeny "AH 16" } <<<
  <<<\parenthesize g g'\harmonic>4\markup { \teeny "AH 17" } <<<
  <<<\parenthesize d' d''\harmonic>2\markup { \teeny "AH 19" } <<<
  %pinched harmonics (PH)
  \pinchedHarmonics
  a''\harmonic>2\startTextSpan
  d''\harmonic>4
  e'\harmonic>4\stopTextSpan
  %tapped harmonics (TH)
  <<<\parenthesize g4 g'\harmonic>4\markup { \teeny "TH 17" } <<<
  <<<\parenthesize a4 a'\harmonic>4\markup { \teeny "TH 19" } <<<
  <<<\parenthesize c'\3 c''\harmonic>2\markup { \teeny "TH 17" } <<<
  %touch harmonics (TCH)
  a4(\harmonicByFret #9 g2.)\markup { \teeny "TCH" } <<<
}

frettedStrings = {
  %artificial harmonics (AH)
  \harmonicByFret #4 g4\3
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 g2\3
  %pinched harmonics (PH)
  \harmonicByFret #7 d2\4
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 a4\5
  %tapped harmonics (TH)
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 d4\4
  \harmonicByFret #5 g2\3
  %touch harmonics (TCH)
  a4 \harmonicByFret #9 g2.\3
}

\score {
  <<
    \new Staff
    \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \harmonics
      }
    }
  \new TabStaff {
    \new TabVoice {
      \frettedStrings
    }
  }
  >>
}
Gleiten *(Glissando)* in Tabulatur

Gleiten kann sowohl in normalem Notensystem als auch in Tabulaturen notiert werden:

```latex
\[\begin{array}{l}
\text{slides} = \{
\text{c'}8\3(\text{\glissando} \text{d'}8\3) \\
\text{c'}8\3\text{\glissando} \text{d'}8\3 \text{\hideNotes} \\
\text{\grace} \text{g16\glissando} \text{\unHideNotes} \\
\text{c'}4\3 \text{\afterGrace} \text{d'}4\3\text{\glissando} \{
\text{\stemDown} \text{\hideNotes} \\
\text{g16} \}
\text{\unHideNotes}
\}\n\end{array}\]
```

```latex
\begin{music}
\score{
\new Staff { \clef "treble_8" \slides }
\new TabStaff { \slides }
\layout {
\context {
\Score
\override Glissando.minimum-length = #4
\override Glissando.springs-and-rods =
\#ly:spanner::set-spacing-rods
\override Glissando.thickness = #2
\omit StringNumber
% or:
%\override StringNumber.stencil = ##f
}
}
\endmusic}
```

Gleiten (Glissando) in Tabulatur
**Akkordglissando in Tabulaturen**


```latex
myMusic = \relative c' {
  <c e g>1 \glissando <f a c>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      myMusic
    }
    \new TabStaff myMusic
  >>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      myMusic
    }
    \new TabStaff \with { \override Glissando.style = #'none } { myMusic }
  >>
}
```

Siehe auch

Notationsreferenz: [Häse], Seite 210, [Akkord-Wiederholungen], Seite 154, [Ausgeschriebene Wiederholungen], Seite 145, [Flageolett], Seite 318, [Glissando], Seite 129.


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:

\begin{verbatim}
melodia = \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
{ \melodia }
>>
\end{verbatim}

Spezialeffekte für Gitarre beschränken sich auf Flageolett und Slide.

Angepasste Tabulaturen

LilyPond errechnet automatisch den Bund für eine Note auf Grundlage der Saite, zu welcher der Ton zugeordnet ist. Um das tun zu können, muss die Stimmung der Saite angegeben werden. Die Stimmung wird in der StringTunings-Eigenschaften bestimmt.

LilyPond hat vordefinierte Stimmungen für Banjo, Mandoline, Gitarre, Bassgitarre, Ukulele, Geige, Bratsche, Cello und Kontrabass. Für diese Stimmungen wird automatisch die richtige Transposition eingesetzt. Das nächste Beispiel ist für Bassgitarre, welche eine Oktave niedriger erklingt, als sie geschrieben ist:

\begin{verbatim}
<<
  \new Voice \with {
    \omit StringNumber
  } {
    \clef "bass_8"
    \relative { c,4 d e f }
  }
  \new TabStaff \with {
    stringTunings = #bass-tuning
  } {
    \relative { c,4 d e f }
  }
\end{verbatim}
Die Standardstimmung ist die Gitarrenstimmung (guitar-tuning) in der EADGHE-Stimmung. Andere vordefinierte Stimmungen sind: guitar-open-g-tuning, mandolin-tuning und banjo-open-g-tuning. Die vordefinierten Stimmungen finden sich in 1y/string-tunings-init.ly.

Jede beliebige Stimmung kann erstellt werden. Die Funktion \stringTuning kann benutzt werden, um eine Saitenstimmung zu definieren und als den Wert von stringTunings für den aktuellen Kontext zu bestimmen.

Als Argument braucht die Funktion eine Akkordkonstruktion, die die Tonhöhen jeder Saiten der Stimmung angibt. Die Akkordkonstruktion muss im absoluten Oktavenmodus angegeben werden, siehe [Absolute Oktavenbezeichnung], Seite 1. Die Saiten mit der höchsten Zahl (normalerweise die tiefste Seite) muss im Akkord zuerst geschrieben werden. Eine Stimmung für ein viersaitiges Instrument mit den Tonhöhen a', d', g' und c' kann folgenderweise erstellt werden:

```latex
mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}
```

Die stringTunings-Eigenschaft wird auch von FretBoards benutzt, um automatische Bund- diagramme zu errechnen.

Saitenstimmungen werden als Teil des Hash-Schlüsselwertes für vordefinierte Bunddiagramme eingesetzt (siehe auch [Vordefinierte Bund-Diagramme], Seite 346.
Das vorherige Beispiel könnte auch folgenderweise geschrieben werden:

```
custom-tuning = \stringTuning <c' g' d'' a''>
```

```
mynotes = {
    c'4 e' g' c'' |
    e''4 g'' b'' c'''
}
```

```
\new Staff {
    \clef treble
    \mynotes
}
\new TabStaff {
    \set TabStaff.stringTunings = #custom-tuning
    \mynotes
}
```

Intern ist die Stimmung eine Scheme-Liste von Tonhöhen der Saiten, eine für jede Saitennummer 1 bis n, wobei 1 die höchste Saiten- und n die unterste. Normalerweise wird so die Stimmung vom höchsten bis zum tiefsten Ton angegeben, aber bei einige Instrumente (etwa Ukulele) werden die Saiten nicht aufgrund der Tonhöhe angeordnet. Die Tonhöhe einer Saite in einer Seitenstimmungsliste ist ein Tonhöhenobjekt für LilyPond. Tonhöhenobjekte werden mit der Scheme-Funktion `+ly:make-pitch` erstellt (siehe Abschnitt A.21 [Scheme-Funktionen], Seite 759).

\`\stringTuning` erstellt derartige Objekte aus der Akkord-Eingabe.

LilyPond errechnet automatisch die Linienanzahl für die Tabulatur und die Zahl der Saiten in dem automatisch erstellten `FretBoard` (Bunddiagramm) aus der Anzahl der Elemente von \`\stringTunings`.

Um für alle `TabStaff`-Kontexte die selbe Standardstimmung zu benutzen, kann man benutzen:

```
\layout {
    \context {
        \TabStaff
        stringTunings = \stringTuning <c' g' d'' a''>
    }
}
```

Auch ein moderner Tabulatur-Schlüssel kann verwendet werden:

```
\new TabStaff {
    \clef moderntab
    <a, e a>1
    \break
```
Der moderne Tabulatur-Schlüssel unterstützt Tabulaturen von 4 bis 7 Saiten.

Siehe auch
Notationsreferenz: [Absolute Oktavenbezeichnung], Seite 1, [Vordefinierte Bund-Diagramme], Seite 346, Abschnitt A.21 [Scheme-Funktionen], Seite 759.
Installierte Dateien: ly/string-tunings-init.ly scm/tablature.scm
Referenz der Interna: Abschnitt “Tab_note_heads_engraver” in Referenz der Interna.

Bekannte Probleme und Warnungen
Automatische Tabulatur-Berechnung funktioniert in den meisten Fällen nicht korrekt bei Instrumenten, deren Saitenstimmung nicht monotonisch fortschreitet, wie etwa Ukulele.

Bund-Diagramm-Beschriftung


```latex
<<
  \context ChordNames {
    \chordmode {
      c1 d:m
    }
  }

  \context Staff {
    \clef "treble_8"
    \context ChordNames {
      \chordmode {
        c1 d:m
      }
    }
    \context Staff {
      \clef "treble_8"
    }
  }
```

Der moderne Tabulatur-Schlüssel unterstützt Tabulaturen von 4 bis 7 Saiten.

Siehe auch
Notationsreferenz: [Absolute Oktavenbezeichnung], Seite 1, [Vordefinierte Bund-Diagramme], Seite 346, Abschnitt A.21 [Scheme-Funktionen], Seite 759.
Installierte Dateien: ly/string-tunings-init.ly scm/tablature.scm
Referenz der Interna: Abschnitt “Tab_note_heads_engraver” in Referenz der Interna.

Bekannte Probleme und Warnungen
Automatische Tabulatur-Berechnung funktioniert in den meisten Fällen nicht korrekt bei Instrumenten, deren Saitenstimmung nicht monotonisch fortschreitet, wie etwa Ukulele.

Bund-Diagramm-Beschriftung


```latex
<<
  \context ChordNames {
    \chordmode {
      c1 d:m
    }
  }

  \context Staff {
    \clef "treble_8"
    \context ChordNames {
      \chordmode {
        c1 d:m
      }
    }
    \context Staff {
      \clef "treble_8"
    }
  }
```
Barré kann hinzugefügt werden:

\context ChordNames {
  \chordmode {
    f1 g
  }
}
\context Staff {
  \clef "treble_8"
  <f, c f a c' f'>1^\markup {
    \fret-diagram "c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
  }
  <g, d g b d' g'>1^\markup {
    \fret-diagram "c:6-1-3;6-3;5-5;4-5;3-4;2-3;1-3;"
  }
}

Die Größe des Bunddiagrammes und die Anzahl der Bünde im Diagramm kann geändert werden:

\context ChordNames {
  \chordmode {
    f1 g
  }
}
\context Staff {
  \clef "treble_8"
  <f, c f a c' f'>1^\markup {
    \fret-diagram "s:1.5;c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
  }
  <g, b, d g b g'>1^\markup {
    \fret-diagram "h:6;6-3;5-2;4-o;3-o;2-o;1-3;"
  }
}

Die Anzahl der Saiten in einem Bunddiagramm kann geändert werden, um sie für andere Instrumente anzupassen, wie etwas Banjo oder Ukulele.

\context ChordNames {
  \chordmode {
    a1
Fingersatz kann auch angezeigt werden, und die Position der Fingersatzzahlen kann kontrolliert werden.

```
<< \context ChordNames {
  \chordmode {
    c1 d:m
  }
}
\context Staff {
  \clef "treble_8"
  <c e g c' e'>1\markup {
    \fret-diagram "f:1;6-x;5-3-3;4-2-2;3-o;2-1-1;1-o;"
  }
  <d a d' f'>1\markup {
    \fret-diagram "f:2;6-x;5-x;4-o;3-2-2;2-3-3;1-1-1;"
  }
}
>>
C\textsuperscript{8} 8 \textsuperscript{10} 10 \textsuperscript{12}  D\textsuperscript{8} 8 \textsuperscript{10} 10 \textsuperscript{12}
```

Die Größe und Position der Punkte kann geändert werden:

```
<< \context ChordNames {
  \chordmode {
    c1 d:m
  }
}
\context Staff {
  \clef "treble_8"
  <c e g c' e'>1\markup {
    \fret-diagram "d:0.35;6-x;5-3-3;4-2-2;3-o;2-1-1-o;"
  }
  <d a d' f'>1\markup {
    \fret-diagram "p:0.2;6-x;5-x;4-o;3-2-2;2-3-1-1;"
  }
}
>>
C\textsuperscript{8} 8 \textsuperscript{10} 10 \textsuperscript{12}  D\textsuperscript{8} 8 \textsuperscript{10} 10 \textsuperscript{12}
```

\[
<< \\texttt{context ChordNames} { \\texttt{chordmode} { \\texttt{c1 d:m} } } >> \\
\texttt{context Staff} { \\texttt{clef "treble_8"} \<c e g c' e'>1\texttt{\markup { \texttt{fret-diagram-terse "x;3;2;o;1;o;"}} } \<d a d' f'>1\texttt{\markup { \texttt{fret-diagram-terse "x;o;2;3;1;"}} } } >>
\]

\texttt{C8 8 10 10 12} \texttt{D8 8 \{10 \}} \texttt{10 12}

Barré kann im knappen Modus auch angezeigt werden:

\[
<< \\texttt{context ChordNames} { \\texttt{chordmode} { \\texttt{f1 g} } } >> \\
\texttt{context Staff} { \\texttt{clef "treble_8"} \<f, c f a c' f'>1\texttt{\markup { \texttt{fret-diagram-terse "1-;3;3;2;1;1-;"}} } \<g, d g b d' g'>1\texttt{\markup { \texttt{fret-diagram-terse "3-;5;5;4;3;3-;"}} } } >>
\]

\texttt{F8 10 12 15 15 17 19} \texttt{G8 10 12 15 15 17 19}

Fingersatz kann im knappen Modus hinzugefügt werden:

\[
<< \\texttt{context ChordNames} { \\texttt{chordmode} { \\texttt{c1 d:m} } } >>
\]

\context Staff {
\override Voice.TextScript.fret-diagram-details.finger-code = #'below-string
\clef "treble_8"
\markup {\fret-diagram-terse "x;3-3;2-2;o;1-1;o;"
\markup {\fret-diagram-terse "x;x;o;2-2;3-3;1-1;"
}
}
}

\context ChordNames {
\chordmode {c1 d:m}
}
\context Staff {
\clef "treble_8"
\markup {\fret-diagram-verbose #'(mute 6)
(place-fret 5 3)
(place-fret 4 2)
(open 3)
(place-fret 2 1)
(open 1)
}
\markup {\fret-diagram-verbose #'(mute 6)
(mute 5)
(open 4)
(place-fret 3 2)
(place-fret 2 3)
(place-fret 1 1)
}
}

Andere Eigenschaften der Bunddiagramme müssen im knappen Modus mit \override-Befehlen angegeben werden.

Die Beschriftungsfunktion \fret-diagram-verbose (ausführlicher Stil) ist in der Form eine Scheme-Liste. Jedes Element stellt ein Element dar, dass im Bunddiagramm gesetzt werden soll.

\context ChordNames {
  \chordmode {
    f1 g c
  }
}
\context Staff {
  \clef "treble_8"
  \override Voice.TextScript.fret-diagram-details.finger-code = #'below-string
  <f, c f a c' f'>1\markup {
    \fret-diagram-verbose #'(
      (place-fret 6 1)
      (place-fret 5 3)
      (place-fret 4 3)
      (place-fret 3 2)
      (place-fret 2 1)
      (place-fret 1 1)
      (barre 6 1 1)
    )
  }
  <g, b, d g b g'>1\markup {
    \fret-diagram-verbose #'(
      (place-fret 6 3 2)
      (place-fret 5 2 1)
      (open 4)
      (open 3)
      (open 2)
      (place-fret 1 3 3)
    )
  }
  <c e g c' e'>1\markup {
    \fret-diagram-verbose #'(
      (capo 3)
      (mute 6)
      (place-fret 4 5 1)
      (place-fret 3 5 2)
      (place-fret 2 5 3)
    )
  }
}

Alle anderen Bunddiagramm-Eigenschaften müssen im ausführlichen Modus mit \override-Befehlen angegeben werden.

Ausgewählte Schnipsel

*Changing fret orientations*

Fret diagrams can be oriented in three ways. By default the top string or fret in the different orientations will be aligned.

```latex
\include "predefined-guitar-fretboards.ly"
<<
\chords {
  c1
  c1
}
\new FretBoards {
  \chordmode {
    c1
    \override FretBoard.fret-diagram-details.orientation = #'landscape
    c1
    \override FretBoard.fret-diagram-details.orientation = #'opposing-landscape
    c1
  }
}
\new Voice {
  c'1
  c'1
  c'
}
>>
```

Anpassung von Beschriftungs-Bunddiagrammen


```latex
<<
\chords { c1 | c | c | d }
\new Voice = "mel" {
```
\textLengthOn
% Set global properties of fret diagram
\override TextScript.size = #'1.2
\override TextScript.fret-diagram-details.finger-code = #'in-dot
\override TextScript.fret-diagram-details.dot-color = #'white

%% C major for guitar, no barre, using defaults
% terse style
c'1\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

%% C major for guitar, barred on third fret
% verbose style
% size 1.0
% roman fret label, finger labels below string, straight barre
c'1\markup {
  % standard size
  \override #'(size . 1.0) {
    \override #'(fret-diagram-details . ( 
      (number-type . roman-lower) 
      (finger-code . in-dot) 
      (barre-type . straight)))) {
    \fret-diagram-verbose #'((mute 6) 
      (place-fret 5 3 1) 
      (place-fret 4 5 2) 
      (place-fret 3 5 3) 
      (place-fret 2 5 4) 
      (place-fret 1 3 1) 
      (barre 5 1 3))
  }
  }
}

%% C major for guitar, barred on third fret
% verbose style
% landscape orientation, arabic numbers, M for mute string
% no barre, fret label down or left, small mute label font
c'1\markup {
  \override #'(fret-diagram-details . ( 
    (finger-code . below-string) 
    (number-type . arabic) 
    (label-dir . -1) 
    (mute-string . "M") 
    (orientation . landscape) 
    (barre-type . none) 
    (xo-font-magnification . 0.4) 
    (xo-padding . 0.3))) {
    \fret-diagram-verbose #'((mute 6) 
      (place-fret 5 3 1) 
      (place-fret 4 5 2) 
      (place-fret 3 5 3) 
      (place-fret 2 5 4) 
      (place-fret 1 3 1)
Kapitel 2: Spezielle Notation

(bare 5 1 3))

%% simple D chord
% terse style
% larger dots, centered dots, fewer frets
% label below string
d'1`\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (dot-radius . 0.35)
    (dot-position . 0.5)
    (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}
}
}
>>

C    C    C    D

Siehe auch
Notationsreferenz: Abschnitt A.10.5 [Instrument Specific Markup], Seite 686.
Referenz der Interna: Abschnitt “fret-diagram-interface” in Referenz der Interna.

Vordefinierte Bund-Diagramme

Bunddiagramme können mit dem FretBoards-Kontext angezeigt werden. Standardmäßig zeigt der FretBoards-Kontext Bunddiagramme an, die in einer Tabelle definiert sind:
\include "predefined-guitar-fretboards.ly"
\context FretBoards {
  \chordmode {
    c1 d
  }
}

Beispiele in dieser Datei können auch für andere Instrumente oder Stimmungen Diagramme definiert werden.

Bunddiagramme für die Ukulele finden sich in der Datei predefined-ukulele-fretboards.ly.

\include "predefined-ukulele-fretboards.ly"

myChords = \chordmode { a1 a:m a:aug }

\new ChordNames {
  \myChords
}

\new FretBoards {
  \set Staff.stringTunings = #ukulele-tuning
  \myChords
}

\new FretBoards {
  \set Staff.stringTunings = #mandolin-tuning
  \myChords
}

Tonhöhen von Akkorden können entweder als Akkordkonstrukte oder im Akkordmodus notiert werden (siehe auch [Überblick über den Akkord-Modus], Seite 388).

\include "predefined-guitar-fretboards.ly"
\context FretBoards {
  \chordmode { c1 }
  <c' e' g'>1
}

\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 f g
}

<<
\context ChordNames {
  mychords
}
\context FretBoards {
  mychords
}
>>

Vordefinierte Bunddiagramme können transponiert werden, solange ein Diagramm für den transponierten Akkord in der Bunddiagramm-Tabelle vorhanden ist.

\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 f g
}

mychordlist = {
  mychords
  \transpose c e { \mychords }
}

<<
\context ChordNames {
  mychordlist
}
\context FretBoards {
  mychordlist
}
>>

Die Tabelle der vordefinierten Bunddiagramme für Gitarre enthält acht Akkorde (Dur, Moll, übermäßig, vermindert, Dominantseptakkord, große Septime, kleine Septime und Dominantnonenakkord) für alle 17 Tonarten. Die Tabelle der vordefinierten Bunddiagramme für


```ly
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
c1 c:maj9
}

<<
\context ChordNames {
  \mychords
}
\context FretBoards {
  \mychords
}
>>
```

C C9


```ly
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
  \chordmode { c:maj9 }
  #guitar-tuning
  "x;3-2;0;0;0;0;"

mychords = \chordmode {
c1 c:maj9
}

<<
\context ChordNames {
  \mychords
}
\context FretBoards {
  \mychords
}
>>
```
Unterschiedliche Bunddiagramme für den selben Akkord können gespeichert werden, indem unterschiedliche Oktaven für die Tonhöhe benutzt werden. Die unterschiedliche Oktave sollte mindestens zwei Oktaven über oder unter der Standardoktave liegen, die für transponierende Bunddiagramme eingesetzt wird.

\include "predefined-guitar-fretboards.ly"
\storePredefinedDiagram #default-fret-table
\chordmode { c'' }
#guitar-tuning
#(offset-fret 2 (chord-shape 'bes guitar-tuning))

mychords = \chordmode {
c1 c''
}

<<
\context ChordNames {
  mychords
}
\context FretBoards {
  mychords
}
>>

Zusätzlich zu Bunddiagrammen speichert LilyPond auch eine interne Liste an Akkordformen. Die Akkordformen sind Bunddiagramme, die am Hals entlang verschoben werden können und dabei unterschiedliche Akkorde ergeben. Akkordformen können zu der internen Liste hinzugefügt werden und dann benutzt werden, um vordefinierte Bunddiagramme zu definieren. Weil sie auf verschiedenen Positionen auf dem Steg gelegt werden können, beinhalten vordefinierte Akkord üblicherweise keine leeren Saiten. Wie Bunddiagramme können auch Akkordformen entweder als \fret-diagram-terse-Definition oder als \fret-diagram-verbose-Liste erstellt werden.

\include "predefined-guitar-fretboards.ly"

% Add a new chord shape
\addChordShape #'powerf #guitar-tuning "1-1;3-3;3-4;x;x;x;"

% add some new chords based on the power chord shape
\storePredefinedDiagram #default-fret-table
\chordmode { f'' }
#guitar-tuning
#(chord-shape 'powerf guitar-tuning)
\storePredefinedDiagram #default-fret-table
\chordmode { g'' }  
#guitar-tuning  
#(offset-fret 2 (chord-shape 'powerf guitar-tuning))  

mychords = \chordmode{ 
  f1 f'' g g'' 
}

<<<  
\context ChordNames { 
  mychords 
}  
\context FretBoards { 
  mychords 
}  
>>>


Ausgewählte Schnipsel

Bunddiagramme anpassen


\include "predefined-guitar-fretboards.ly"
\storePredefinedDiagram #default-fret-table \chordmode { c' }  
#guitar-tuning  
#"x;1-1-(;3-2;3-3;3-4;1-1-);"  

<<<  
\new ChordNames { 
  \chordmode { cl | c | c | d } 
}  
\new FretBoards {  
  % Set global properties of fret diagram  
  \override FretBoards.FretBoard.size = #'1.2  
  \override FretBoard.fret-diagram-details.finger-code = #'in-dot  
  \override FretBoard.fret-diagram-details.dot-color = #'white  
  \chordmode {  
    c  
    \once \override FretBoard.size = #'1.0  
    \once \override FretBoard.fret-diagram-details.barre-type = #'straight  
    \once \override FretBoard.fret-diagram-details.dot-color = #'black
Eigene vordefinierte Bunddiagramme für andere Instrumente erstellen

Vordefinierte Bunddiagramme können für neue Instrumente hinzugefügt werden neben denen, die schon für die Gitarre definiert sind. Dieses Schnipsel zeigt, wie man eine neue Saitenstimung definiert und dann eigene vordefinierte Bunddiagramme bestimmt. Das Beispiel ist für das venezuelische Cuatro.

Dieses Schnipsel zeigt auch, wie Fingersatz in die Akkorde eingebunden werden kann, um als Referenzpunkt für die Akkordauswahl benutzt werden kann. Dieser Fingersatz wird im Bunddiagramm und in der Tabulatur, aber nicht in den Noten angezeigt.

Diese Bunddiagramme sind nicht transponierbar, weil sie Saiteninformationen enthalten. Das soll in der Zukunft verbessert werden.

\% add FretBoards for the Cuatro
\% Note: This section could be put into a separate file
\% predefined-cuatro-fretboards.ly
\% and \included into each of your compositions

cuatroTuning = #\(,(ly:make-pitch 0 6 0)
    ,(ly:make-pitch 1 3 SHARP)
    ,(ly:make-pitch 1 1 0)
    ,(ly:make-pitch 0 5 0))\)
dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }
aMajSeven = { <a\4 cis\1 e\3 g\2> }
dMajSeven = \{ <a\4 c\1 d\3 fis\2> \}
gMajor = \{ <b\4 b\1 d\3 g\2> \}

\storePredefinedDiagram #default-fret-table \dSix
  #cuatroTuning
  "o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
  #cuatroTuning
  "o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
  #cuatroTuning
  "o;2-2;1-1;2-3;"
\storePredefinedDiagram #default-fret-table \dMajSeven
  #cuatroTuning
  "o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
  #cuatroTuning
  "2-2;o;1-1;o;"

% end of potential include file /predefined-cuatro-fretboards.ly

%(set-global-staff-size 16)

primerosNames = \chordmode {
d:6 d a:maj7 d:maj7
g}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c' {
        \primeros
      }
    }

    \new FretBoards {
      \set Staff.stringTunings = #cuatroTuning
      % \override FretBoard
% #'(fret-diagram-details string-count) = #'4
\override FretBoard.fret-diagram-details.finger-code = #'in-dot
\primeros
\}
\new TabStaff \relative c'' {
\set TabStaff.stringTunings = #cuatroTuning
\primeros
\}
\layout {
\context {
\Score
\override SpacingSpanner.base-shortest-duration = #(ly:make-moment 1 16)
\}
\midi { }
\}

\include "predefined-guitar-fretboards.ly"

myChords = \chordmode {
c1 c1 \break
\set chordChanges = ##t
c1 c1 \break
c1 c1
}
<<
\new ChordNames { \myChords }
\new FretBoards { \myChords }
\new Staff { \myChords }
>>

Akkordänderungen für Bunddiagramme

Bunddiagramme können definiert werden, sodass sie nur angezeigt werden, wenn der Akkord sich ändert oder eine neue Zeile anfängt.
Alternative Bunddiagrammtabellen

Alternative Bunddiagrammtabellen können erstellt werden. Sie können benutzt werden, um alternative Bunddiagramme für einen bestimmten Akkord zu haben.

Damit eine alternative Bunddiagrammentabelle benutzt werden kann, muss die Tabelle zuerst erstellt werden. Dann werden die Bunddiagramme zur Tabelle hinzugefügt.

Die erstellte Bunddiagrammtabelle kann auch leer sein, oder sie kann aus einer existierenden Tabelle kopiert werden.

Die Tabelle, die eingesetzt wird, um vordefinierte Bunddiagramme anzuzeigen, wird mit der Eigenschaft \texttt{\predefinedDiagramTable} ausgewählt.

\texttt{\include "predefined-guitar-fretboards.ly"}

% Make a blank new fretboard table
#(define custom-fretboard-table-one (make-fretboard-table))

% Make a new fretboard table as a copy of default-fret-table
#(define custom-fretboard-table-two (make-fretboard-table default-fret-table))

% Add a chord to custom-fretboard-table-one
\storePredefinedDiagram #custom-fretboard-table-one
  \chordmode{c}
  \guitar-tuning
  "3-(;3;5;5;5;3-);"

% Add a chord to custom-fretboard-table-two
\storePredefinedDiagram #custom-fretboard-table-two
  \chordmode{c}
  \guitar-tuning
  "x;3;5;5;5;o;"

<<
  \chords {
Siehe auch

Notationsreferenz: [Angepasste Tabulaturen], Seite 335, [Automatische Bund-Diagramme], Seite 357, [Überblick über den Akkord-Modus], Seite 388, Abschnitt A.4 [Die vordefinierten Bund-Diagramme], Seite 613.

Installierte Dateien: ly/predefined-guitar-fretboards.ly,
ly/predefined-guitar-ninth-fretboards.ly,
ly/predefined-ukulele-fretboards.ly,
ly/predefined-mandolin-fretboards.ly.


Referenz der Interna: Abschnitt “fret-diagram-interface” in Referenz der Interna.
Automatische Bund-Diagramme

Bunddiagramme können automatisch aus notierten Noten erstellt werden. Hierzu wird der FretBoards-Kontext eingesetzt. Wenn keine vordefinierten Diagramme für die entsprechenden Noten mit der aktiven Saitenstimmung (stringTunings) vorhanden sind, errechnet der Kontext Saiten und Bünde die benutzt werden können, um die Noten zu spielen.

\context ChordNames {
  \chordmode {
    f1 g
  }
}\context FretBoards {
  <f, c f a c' f'>1
  <g, 6 b, d g b g'>1
}\context Staff {
  \clef "treble_8"
  <f, c f a c' f'>1
  <g, b, d g b' g'>1
}>}

\storePredefinedDiagram #default-fret-table
  <c e g c' e'>
  #guitar-tuning
  "x;3-1-(;5-2;5-3;5-4;3-1-1-);"
<<
\context ChordNames {
  \chordmode {
    cl c c
  }
}\context FretBoards {
  <c e g c' e'>1
  \predefinedFretboardsOff
  <c e g c' e'>1
  \predefinedFretboardsOn
  <c e g c' e'>1
}\context Staff {
  \clef "treble_8"
  <c e g c' e'>1
  <c e g c' e'>1
>>

Da in den Standardeinstellungen keine vordefinierten Diagramme geladen werden, ist die automatische Diagrammerstellung das Standardverhalten. Wenn die vordefinierten Diagramme eingesetzt werden, kann die automatische Berechnung an- und ausgeschaltet werden.

\storePredefinedDiagram #default-fret-table
  <c e g c' e'>
  #guitar-tuning
  "x;3-1-(;5-2;5-3;5-4;3-1-1-);"
Manchmal kann die Berechnungsfunktion für Bunddiagramme kein passendes Diagramm finden. Das kann oft umgangen werden, indem man manuell einer Note eine bestimmte Saiten zuweist. In vielen Fällen muss nur eine Note derart gekennzeichnet werden, der Rest wird dann entsprechend durch den FretBoards-Kontext behandelt.

Fingersatz kann zu FretBoard-Bunddiagrammen hinzugefügt werden.

\<<
\context ChordNames {
  \chordmode {
    c1 d:m
  }
}\context FretBoards {
  <c-3 e-2 g c'-1 e'>1
  <d a-2 d'-3 f'-1>1
}\context Staff {
  \clef "treble_8"
  <c e g c' e'>1
  <d a d' f'>1
}\>>

Der kleinste Bund, der benutzt wird, um Saiten und Bünde im FretBoard-Kontext zu errechnen, kann mit der minimumFret-Eigenschaft gesetzt werden.

\<<
\context ChordNames {
  \chordmode {
    d1:m d:m
  }
}\context FretBoards {
  <d a d' f'>1
  \set FretBoards.minimumFret = #5
  <d a d' f'>1
}\context Staff {
  \clef "treble_8"
  <c e g c' e'>1
  <d a d' f'>1
  \clef "treble_8"
}
Die Saiten und Bünde des FretBoards-Kontextes hängen von der stringTunings-
Eigenschaft ab, die die gleiche Bedeutung wie im TabStaff-Kontext hat. Siehe auch [Angepasste
Tabulaturen], Seite 335, zu Information über die stringTunings-
Eigenschaft.

Die graphische Erscheinung eines Bunddiagrammes kann den Bedürfnissen angepasst werden,
indem Eigenschaften der fret-diagram-interface-Schnittstelle verändert werden. Einzelheiten
finden sich in Abschnitt “fret-diagram-interface” in Referenz der Interna. Die Schnittstel-
leneigenschaften eines FretBoards-Diagramms gehören dem FretBoards.FretBoard-Kontext
an.

Vordefinierte Befehle
\predefinedFretboardsOff, \predefinedFretboardsOn.

Siehe auch

Notationsreferenz: [Angepasste Tabulaturen], Seite 335.
Referenz der Interna: Abschnitt “fret-diagram-interface” in Referenz der Interna.

Bekannte Probleme und Warnungen
Autogeneische Bundberechnung funktioniert nicht richtig für Instrumente mit nicht-
monotonischer Stimmung.

Fingersatz der rechten Hand
cindex Fingersatz der rechten Hand, Bundinstrumente

Fingersatz für die rechte Hand p-i-m-a muss mit \rightHandFinger, gefolgt von einer Zahl,
notiert werden.

\Achtung: Wenn die Zahl in Scheme-Notation eingegeben wird, muss da-
rauf geachtet werden, ein Leerzeichen anzuhängen vor dem schließenden
> o. Ä.

\clef "treble_8"
c4[rightHandFinger #1
e[rightHandFinger #2
g[rightHandFinger #3
c'[rightHandFinger #4<
c[rightHandFinger #1 e[rightHandFinger #2
\g[rightHandFinger #3 c'[rightHandFinger #4 >1

\clef "treble_8"
c4[rightHandFinger #1
e[rightHandFinger #2
g[rightHandFinger #3
c'[rightHandFinger #4<
c[rightHandFinger #1 e[rightHandFinger #2
\g[rightHandFinger #3 c'[rightHandFinger #4 >1
Zur Erleichterung kann der Befehl \rightHandFinger zu ein paar Buchstaben abgekürzt werden, etwa RH.

RH=#rightHandFinger

**Ausgewählte Schnipsel**

*Positionierung von Fingersatz der rechten Hand*

Man kann die Positionierung von Fingersatz der rechten Hand besser kontrollieren, wenn eine bestimmte Eigenschaft gesetzt wird, wie das folgende Beispiel zeigt:

```latex
#(define RH rightHandFinger)

\relative c {
\clef "treble_8"
\set strokeFingerOrientations = #'(up down)
\clef "treble_8"
\set strokeFingerOrientations = #'(up right down)
\clef "treble_8"
\set strokeFingerOrientations = #'(left)
\}
```

*Fingersatz, Saitennummern und Fingersatz für die rechte Hand*

Dieses Beispiel kombiniert Fingersatz für die linke Hand, Saitennummern und Fingersatz für die rechte Hand.

```latex
#(define RH rightHandFinger)

\relative c {
\clef "treble_8"
\set strokeFingerOrientations = #'(left)
\}
```

Siehe auch


Referenz der Interna: Abschnitt “StrokeFinger” in Referenz der Interna.
2.4.2 Gitarre

Die meisten der Besonderheiten von Gitarrennotation wurden im allgemeinen Abschnitt behandelt, aber es gibt noch einige, die hier gezeigt werden sollen. Teilweise soll ein Lead-sheet nur die Akkordsymbole und den Gesangstext enthalten. Da LilyPond ein Notensatzprogramm ist, wird es nicht für derartige Projekte empfohlen, die keine eigentliche Notation beinhalten. Anstatt dessen sollte ein Textbearbeitungsprogramm oder ein Satzprogramm wie GuitarTeX (für erfahrende Benutzer) eingesetzt werden.

Position und Barré anzeigen

Das Beispiel zeigt, wie man Griff- und Barréposition notieren kann.

```
\relative {
  \clef "treble_8"
  b,16 d g b e
  \textSpannerDown
  \override TextSpanner.bound-details.left.text = "XII "
  g16\startTextSpan
  b16 e g e b g\stopTextSpan
  e16 b g d
}
```

Siehe auch

Notationsreferenz: [Text mit Verbindungslinien], Seite 217.

Flageolett und gedämpfte Noten

Besondere Notenköpfe können eingesetzt werden, um gedämpfte Noten oder Flageolettöne anzuzeigen. Flageolettöne werden normalerweise mit einem Text erklärt.

```
\relative {
  \clef "treble_8"
  \override Staff.NoteHead.style = #'harmonic-mixed
  d'\markup { \italic { \fontsize #-2 { "harm. 12" }}}} <g b>1
}
```

Gedämpfte oder gestoppte Noten werden in normalen und Tabulatur-Systemen unterstützt:

```
music = \relative {
  < a\3 \deadNote c\2 a'\1 >4
  < b\3 \deadNote d\2 b'\1 >
  < c\3 \deadNote e\2 c'\1 >
```
Eine andere Spieltechnik (insbesondere bei elektrischen Gitarren benutzt) ist \textit{palm mute}. Hierbei wird die Saite teilweise durch die Handfläche der Schlaghand gedämpft. LilyPond unterstützt die Notation dieser Art von Technik, indem die Notenköpfen der so gedämmten Noten durch Dreiecke ersetzt werden.

\begin{verbatim}
\new Voice { % Warning: explicit Voice instantiation is
  % required to have \texttt{palmMuteOff} work properly
  % when \texttt{palmMuteOn} comes at the beginning of
  % the piece.
 relative c, { 
  \clef "G_8"
  \palmMuteOn
  e8 \markup { \musicglyph "noteheads.u2do" = palm mute } < e b' e > e
  \palmMuteOff
  e e \palmMute e e e | e e e e |
  e8 \palmMute { e e e } e e e e |
  < \palmMute e b' e >8 \palmMute { e e e } < \palmMute e b' e >2
 }
\end{verbatim}

\begin{verbatim}
\new Voice { % Warning: explicit Voice instantiation is
  % required to have \texttt{palmMuteOff} work properly
  % when \texttt{palmMuteOn} comes at the beginning of
  % the piece.
 relative c, { 
  \clef "G_8"
  \palmMuteOn
  e8 \markup { \musicglyph "noteheads.u2do" = palm mute } < e b' e > e
  \palmMuteOff
  e e \palmMute e e e | e e e e |
  e8 \palmMute { e e e } e e e e |
  < \palmMute e b' e >8 \palmMute { e e e } < \palmMute e b' e >2
 }
\end{verbatim}

\begin{verbatim}
\new Voice { % Warning: explicit Voice instantiation is
  % required to have \texttt{palmMuteOff} work properly
  % when \texttt{palmMuteOn} comes at the beginning of
  % the piece.
 relative c, { 
  \clef "G_8"
  \palmMuteOn
  e8 \markup { \musicglyph "noteheads.u2do" = palm mute } < e b' e > e
  \palmMuteOff
  e e \palmMute e e e | e e e e |
  e8 \palmMute { e e e } e e e e |
  < \palmMute e b' e >8 \palmMute { e e e } < \palmMute e b' e >2
 }
\end{verbatim}
Siehe auch

Notationsreferenz: [Besondere Notenköpfe], Seite 34, Abschnitt A.9 [Notenkopfstile], Seite 646.

Schnipsel: Abschnitt "Fretted strings" in Schnipsel.

**Powerakkorde anzeigen**

Powerakkorde und ihre Symbole können im Akkordmodus oder als Akkordkonstruktionen gesetzt werden:

```latex
ChordsAndSymbols = {
  \chordmode {
    \powerChords
e,,1:5
  a,,1:5.8
  \set minimumFret = #8
c,1:5
  f,1:5.8
  }
  \set minimumFret = #5
  <a, e>1
  <g d' g'>1
}
\score {
  \new ChordNames {
    \ChordsAndSymbols
  }
  \new Staff {
    \clef "treble_8"
    \ChordsAndSymbols
  }
  \new TabStaff {
    \ChordsAndSymbols
  }
  \timesig.C44

  \new ChordNames {
    \ChordsAndSymbols
  }
  \new Staff {
    \clef "treble_8"
    \ChordsAndSymbols
  }
}
```

Powerakkord-Symbole werden automatisch ausgeschaltet, wenn einer der anderen normalen Akkord-Modifikatoren verwendet wird:

```latex
mixedChords = \chordmode {
  c,1
  \powerChords
```
Kapitel 2: Spezielle Notation

Siehe auch

Glossar: Abschnitt “power chord” in Glossar.

Notationsreferenz: [Erweiterte und modifizierte Akkorde], Seite 390, [Akkordbezeichnungen drucken], Seite 393.


2.4.3 Banjo

Banjo-Tabulaturen

LilyPond hat grundlegende Unterstützung für 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 fünsaitiges Banjo sind in LilyPond vordefiniert: banjo-c-tuning (gCGBD), banjo-modal-tuning (gDGCD), banjo-open-d-tuning (aDF#AD) und banjo-open-dm-tuning (aDFAD).

Sie 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

Installierte Dateien: ly/string-tunings-init.ly.

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.

Referenz für Schlagzeug

- Viele Schlagzeugmusik kann auf einem rhythmischen System notiert werden. Das wird gezeigt in [Melodierhythmus anzeigen], Seite 75, und [Neue Notensysteme erstellen], Seite 174.
- MIDI-Ausgabe wird behandelt in Abschnitt 3.5.6 [Schlagzeug in MIDI], Seite 495.

Siehe auch

Notationsreferenz: [Melodierhythmus anzeigen], Seite 75, [Neue Notensysteme erstellen], Seite 174. Abschnitt 3.5.6 [Schlagzeug in MIDI], Seite 495.
Schnipsel: Abschnitt “Percussion” in Schnipsel.

Grundlagen der Schlagzeugnotation

Schlagzeug-Noten können im \drummode-Modus notiert werden, der sich ähnlich verhält wie der Standardmodus für die Noteneingabe. Am einfachsten kann der \drums-Befehl benutzt werden, der sich um den richtigen Kontext und Eingabemodus kümmert:

\drums {
   hihat4 hh bassdrum bd
}

Das ist die Kurzschreibweise für:

\new DrumStaff {
   \drummode {
      hihat4 hh bassdrum bd
   }
}

Grundlagen der Schlagzeugnotation

Beachten Sie, dass normale Tonhöhen (wie cis4) in einem DrumStaff-Kontext eine Fehlermeldung erzielen. Schlüssel für Schlagzeug werden automatisch hinzugefügt, aber sie können auch explizit gesetzt werden. Auch andere Schlüssel können benutzt werden.

```
\drums {
    \clef percussion
    bd4 bd bd bd
    \clef treble
    hh4 hh hh hh
}
```

Es gibt einige Probleme mit der MIDI-Unterstützung für Schlagzeuginstrumente. Details finden sich in Abschnitt 3.5.6 [Schlagzeug in MIDI], Seite 495.

Siehe auch

Notationsreferenz: Abschnitt 3.5.6 [Schlagzeug in MIDI], Seite 495, Abschnitt A.14 [Schlagzeugnoten], Seite 706.


Schnipsel: Abschnitt “Percussion” in Schnipsel.

Trommelwirbel


```
\drums {
    \time 2/4
    sn16 sn8 sn16 sn8 sn8:32 ~
    sn8 sn8 sn4:32 ~
    sn4 sn8 sn16 sn16
    sn4 r4
}
```

Benutzung der Stöcke kann angezeigt werden, indem eine Beschriftung durch "R" oder "L" über oder unter der Note angefügt wird, Näheres in Abschnitt 5.4.2 [Richtung und Platzierung], Seite 581. Die staff-padding-Eigenschaft kann verändert werden, um eine Orientierung an einer gemeinsamen Linie zu ermöglichen.

```
\drums {
    \repeat unfold 2 {
        sn16^"L" sn^"R" sn^"L" sn^"L" sn^"R" sn^"L" sn^"R" sn^"L" sn^"R"
    \stemUp
```
Siehe auch

Notationsreferenz: [Tremolo-Wiederholung], Seite 149.
Schnipsel: Abschnitt “Percussion” in Schnipsel.

Schlagzeug mit Tonhöhe

Bestimmte Schlagzeuginstrumente mit Tonhöhe (z. B. Xylophone, vibraphone und Pauken) werden auf normalen Systemen geschrieben. Das wird in anderen Abschnitten des Handbuchs behandelt.

Siehe auch

Notationsreferenz: Abschnitt 3.5.6 [Schlagzeug in MIDI], Seite 495.
Schnipsel: Abschnitt “Percussion” in Schnipsel.

Schlagzeugsysteme


\new DrumStaff <<
\drummode {
  \repeat unfold 16 hh16
}\new DrumStaff {
}\new DrumVoice { \voiceOne \up }
\new DrumVoice { \voiceTwo \down }
>>

Das Beispiel zeigt ausdrücklich definierte mehrstimmige Notation. Die Kurznotation für mehrstimmige Musik, wie sie im Abschnitt Abschnitt “Ich höre Stimmen” in Handbuch zum Lernen beschrieben wird, kann auch verwendet werden.
Es gibt auch weitere Layout-Einstellungen. Um diese zu verwenden, muss die Eigenschaft `drumStyleTable` im DrumVoice-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 `tomfl` 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.

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

Eigene Schlagzeugsysteme
Wenn ihnen keine der vordefinierten Stile gefällt, können Sie auch eine eigene Liste der Positionen und Notenköpfe am Anfang ihrer Datei erstellen.
% This defines a staff with only two lines.
% It also defines the positions of the two lines.
\override Staff.StaffSymbol.line-positions = #'(-2 3)

% This is necessary; if not entered, the barline would be too short!
\override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

\new DrumStaff {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \set DrumStaff.instrumentName = "Tambourine"
  \new Tambourine staff

  \new DrumStaff {
    \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

    \drummode {
      \time 2/4
      wb18 wb116 wb1 wbh8-> wbl |
      wbl8 wb116 wbh-> ~ wbl wb116 r8 |
    }
  }
}

\new DrumStaff {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

  \drummode {
    \time 6/8
    tamb8. tamb16 tamb8 tamb tamb tamb |
    tamb4. tamb8 tamb tamb |
    \% the trick with the scaled duration and the shorter rest
    \% is neccessary for the correct ending of the trill-span!
    tamb2.*5/6 \startTrillSpan s8 \stopTrillSpan |
  }
}
Kapitel 2: Spezielle Notation

Tambourine

Noten für Tam-Tam (notiert mit tt):

```lilypond
#(define mydrums '((tamtam default #t 0)))

tamtamstaff = {
    \override Staff.StaffSymbol.line-positions = #'( 0 )
    \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)
    \set DrumStaff.instrumentName = "Tamtam"
}

\new DrumStaff {
    \tamtamstaff
    \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)

    \drummode {
        tt 1 \pp \laissezVibrer
    }
}

Tamtam
```

Zwei Glocken, notiert mit cb (Kuhglocke) und rb (Reiterglocke)

```lilypond
#(define mydrums '((ridebell default #t 3)
    (cowbell default #t -2)))

bellstaff = {
    \override DrumStaff.StaffSymbol.line-positions = #'(-2 3)
    \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
    \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)
    \set DrumStaff.instrumentName = "Different Bells"
}

\new DrumStaff {
    \bellstaff
    \drummode {
        \time 2/4
        rb8 rb cb cb16 rb-> ~ |
        16 rb8 rb16 cb8 cb |
    }
}

Different Bells
```

Hier ein kurzes Beispiel von Stravinsky (aus „L’histoire du Soldat“):

```lilypond
#(define mydrums '((bassdrum default #t 4)
    (snare default #t -4)
    (tambourine default #t 0)))
```
global = {
    \time 3/8 s4.
    \time 2/4 s2*2
    \time 3/8 s4.
    \time 2/4 s2
}

drumsA = {
    \context DrumVoice <<
        \{ \global }
        \{ \drummode {
            \autoBeamOff
            \stemDown sn8 \stemUp tamb s8 |
            sn4 \stemDown sn4 |
            \stemUp tamb8 \stemDown sn8 \stemUp sn16 \stemDown sn \stemUp sn8 |
            \stemDown sn8 \stemUp tamb s8 |
            \stemUp sn4 s8 \stemUp tamb
        }\}
    >>
}

drumsB = {
    \drummode {
        s4 bd8 s2*2 s4 bd8 s4 bd8 s8
    }
}

\layout {
    indent = #40
}

\score {
    \new StaffGroup <<
        \new DrumStaff {
            \set DrumStaff.instrumentName = \markup {
                \column {
                    "Tambourine"
                    "et"
                    "caisse claire s. timbre"
                }
            }
            \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
            \drumsA
        }

        \new DrumStaff {
            \set DrumStaff.instrumentName = "Grosse Caisse"
            \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
            \drumsB
        }
    >>
}
Tambourine
et
caisse claire s. timbre
Grosse Caisse

Siehe auch
Schnipsel: Abschnitt “Percussion” in Schnipsel.


Geisternoten
Geisternoten für Schlagzeug und Perkussion können mit dem Klammer-(\parenthesize)-Befehl, beschrieben in [Klammern], Seite 210, erstellt werden. Im Standard-\drummode-Modus ist aber das \parenthesis_ engraver-Plugin nicht automatisch enthalten.

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

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

Siehe auch
Schnipsel: Abschnitt “Percussion” in Schnipsel.
2.6 Blasinstrumente

Dieser Abschnitt beinhaltet einige Notationselemente, die bei der Notation von Blasinstrumen- ten auftreten.

2.6.1 Übliche Notation für Bläser

Dieser Abschnitt erklärt Notation, die für die meisten Blasinstrumente gültig sind.

Referenz für Blasinstrumente

Viele Besonderheiten der Blasinstrumentenotation haben mit Atmung und Spielart zu tun:

- Atmung kann durch Pausen oder mit Atemzeichen angezeigt werden, siehe [Atemzeichen], Seite 127.
- Legato kann durch Legatobögen angezeigt werden, siehe [Legatobögen], Seite 123.
- Flatterzunge wird angezeigt, indem ein Tremolozeichen und eine Anmerkung für die entsprechende Note gesetzt wird. Siehe [Tremolo-Wiederholung], Seite 149.

Es gibt auch noch weitere Aspekte der Notation, die für Blasinstrumente relevant sein können:

- Viele Instrumente sind transponierend, siehe [Transposition von Instrumenten], Seite 23.
- Das Zug-Glissando ist charakteristisch für die Posaune, aber auch andere Instrumente können Glissandos ausführen. Siehe [Glissando], Seite 129.
- Obertonreihenglissandi, die auf allen Blechblasinstrumenten möglich, aber besonders üblich für das Waldhorn sind, werden üblicherweise mit Verzierungnoten geschrieben. Siehe [Verzierungen], Seite 105.
- Tonhöenschwankungen am Ende eines Tons werden gezeigt in [Glissando zu unbestimmter Tonhöhe], Seite 128.
- Ventil- oder Klappenschläge werden oft als Kreuznoten dargestellt, siehe [Besondere Notenköpfe], Seite 34.
- Die Benutzung von Dämpfern für Blechblasinstrumente wird meistens durch Text gefordert, aber bei schnellem Wechsel bietet es sich an, die Artikulationszeichen stopped und open zu benutzen. Siehe [Artikulationszeichen und Verzierungen], Seite 113, und Abschnitt A.13 [Liste der Artikulationszeichen], Seite 705.
Ausgewählte Schnipsel

\flageolet-Zeichen verkleinern

Um den \flageolet-Kreis kleiner zu machen, kann diese Scheme-Funktion eingesetzt werden.

\begin{verbatim}
smallFlageolet =
    #(let ((m (make-articulation "flageolet")))
        (set! (ly:music-property m 'tweaks)
            (acons 'font-size -3
                (ly:music-property m 'tweaks)))
        m)
\end{verbatim}

\layout { ragged-right = ##f }

\relative c'' {
    d4\flageolet \markup { default size } d\flageolet
    c4\smallFlageolet \markup { smaller } c\smallFlageolet
}

Siehe auch

Notationsreferenz: [Atemzeichen], Seite 127, [Legatobögen], Seite 123, [Artikulationszeichen und Verzierungen], Seite 113, Abschnitt A.13 [Liste der Artikulationszeichen], Seite 705, [Tremolo-Wiederholung], Seite 149, [Transposition von Instrumenten], Seite 23, [Glissando], Seite 129, [Verzierungen], Seite 105, [Glissando zu unbestimmter Tonhöhe], Seite 128, [Besondere Notenköpfe], Seite 34,


Fingersatz

Alle Blasinstrumente außer der Posaune benötigen mehrere Finger, um verschiedene Tonhöhen zu produzieren. Einige Fingersatzbeispiele zeigen die Schnipsel unten.

Diagramme für Holzbläser können erstellt werden nach den Anweisungen in Abschnitt 2.6.3.1 [Holzbläserdiagramme], Seite 379.

Ausgewählte Schnipsel

Fingering symbols for wind instruments

Special symbols can be achieved by combining existing glyphs, which is useful for wind instruments.

\begin{verbatim}
centermarkup = {
    \once \override TextScript.self-alignment-X = #CENTER
    \once \override TextScript.X-offset =#(lambda (g)
        (+ (ly:self-alignment-interface::centered-on-x-parent g)
            (ly:self-alignment-interface::x-aligned-on-self g)))
}
\score {
    \relative c'{
        g\open
}
\end{verbatim}
Recorder fingering chart

The following example demonstrates how fingering charts for wind instruments can be realized.

% range chart for paetzold contrabass recorder

centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset =#(lambda (g)
    (+ (ly:self-alignment-interface::centered-on-x-parent g)
      (ly:self-alignment-interface::x-aligned-on-self g)))
}

\score {
  \new Staff \with {
    \remove "Time_signature_engraver"
    \omit Stem
    \omit Flag
    \consists "Horizontal_bracket_engraver"
  }
  {\clef bass
    \set Score.timing = ##f
    f,1*1/4 \glissando
    \clef violin
    gis'1*1/4
    \stemDown a'4\markup{1}
  }\centermarkup
  \once \override TextScript.padding = #2
  bes'1*1/4\markup{\override #'(baseline-skip . 1.7) \column
    { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2 \finger 3 \finger 4}
Siehe auch

Notationsreferenz: Abschnitt 2.6.3.1 [Holzbläserdiagramme], Seite 379.


2.6.2 Dudelsack

Dieser Abschnitt beinhaltet die Notation von Dudelsackmusik.

Dudelsack-Definitionen

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] }

tzu 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 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.

Einige moderne Dudelsacknoten benutzen halbe Finger auf c und f, um diese Noten zu erniedrigen. Das kann angezeigt werden mit `c-flat` bzw. `f-flat`. Gleicherweise kann das piobaireachd hohe g als `g-flat` geschrieben werden, wenn es in leichter Musik vorkommt.

Siehe auch


Dudelsack-Beispiele

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

```latex
#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 \hdblff 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 "|."}
```
Kapitel 2: Spezielle Notation

Siehe auch

2.6.3 Holzbläser
Dieser Abschnitt zeigt Notation, die spezifisch für Holzbläser ist.

2.6.3.1 Holzbläserdiagramme
Holzbläserdiagramme können benutzt werden, um die Griffe für eine bestimmte Note darzustellen. Diagramme gibt es für folgende Instrumente:

- Piccolo
- Flöte
- Oboe
- Clarinette
- BassClarinette
- Saxophon
- Fagott
- Kontrafagott

Holzbläserdiagramme werden als Beschriftung erstellt:

```latex
\markup {
  c' '1
  \woodwind-diagram #'piccolo #'(lh . (gis))
  (cc . (one three))
  (rh . (ees))
}
```

Amazing Grace

Hymn

Trad. arr.
Löcher können offen, halboffen, Ring oder geschlossen sein:

```
\textLengthOn
\markup {\text{c''1}}
\center-column {
"Ein Viertel"
\woodwind-diagram #'flute '
(cc . (one1q))
(lh . ())
(rh . ())
}
}
\markup {\text{c''1}}
\center-column {
"Eineinhalb"
\woodwind-diagram #'flute '
(cc . (one1h))
(lh . ())
(rh . ())
}
}
\markup {\text{c''1}}
\center-column {
"Drei Viertel"
\woodwind-diagram #'flute '
(cc . (one3q))
(lh . ())
(rh . ())
}
}
\markup {\text{c''1}}
\center-column {
"Ring"
\woodwind-diagram #'flute '
(cc . (oneR))
(lh . ())
(rh . ())
}
}
\markup {\text{c''1}}
\center-column {
"ganz"
\woodwind-diagram #'flute '
(cc . (oneF two))
(lh . ())
(rh . ())
}
}
```
Ein Viertel    Einerthalb    Drei Viertel    Ring    ganz

Triller werden als schattierte Löcher in den Diagrammen angezeigt:

```
c''1\markup {
  \textLengthOn 
  \center-column {
    "ein Viertel zu Ring"
    \woodwind-diagram #'flute #'((cc . (one1qTR))
      (1h . ())
      (rh . ()))
  }
}
```

```
c''1\markup {
  \center-column {
    "Ring zu geschlossen"
    \woodwind-diagram #'flute #'((cc . (oneTR))
      (1h . ())
      (rh . ()))
  }
}
```

```
c''1\markup {
  \center-column {
    "Ring zu geöffnet"
    \woodwind-diagram #'flute #'((cc . (oneRT))
      (1h . ()))
  }
}
```
Die Liste aller möglichen Löcher und Einstellungen eines bestimmten Instruments kann auf der Kommandozeile mit dem Befehl #(print-keys-verbose 'flute) oder in einer Log-Datei mit #(print-keys-verbose 'flute (current-error-port)) angezeigt werden; durch diese Befehle wird der Notensatz nicht verändert.

Ausgewählte Schnipsel

Liste der Holzbläserdiagramme

Folgende Noten zeige alle Holzbläserdiagramme, die für LilyPond definiert sind.

\layout {
    indent = 0
}
\relative c' {
    \textLengthOn
    \markup {
        \center-column { 
            'tin-whistle
            " "
            \woodwind-diagram
            #'tin-whistle
            #'()
        }
    }
    \markup {
        \center-column { 
            'piccolo
            " "
            \woodwind-diagram
            #'piccolo
            #'()
        }
    }
    \markup {
        \center-column { 
            'flute
            " "
            \woodwind-diagram
            #'flute
            #'()
        }
    }
    \markup {
        \center-column { 
            'oboe
            " "
            \woodwind-diagram
            #'oboe
            #'()
        }
    }
}
c1^\markup {
  \center-column {
    'clarinet
    " "
    \woodwind-diagram
    #'clarinet
    #'()
  }
}

c1^\markup {
  \center-column {
    'bass-clarinet
    " "
    \woodwind-diagram
    #'bass-clarinet
    #'()
  }
}

c1^\markup {
  \center-column {
    'saxophone
    " "
    \woodwind-diagram
    #'saxophone
    #'()
  }
}

c1^\markup {
  \center-column {
    'bassoon
    " "
    \woodwind-diagram
    #'bassoon
    #'()
  }
}

c1^\markup {
  \center-column {
    'contrabassoon
    " "
    \woodwind-diagram
    #'contrabassoon
    #'()
  }
}
Graphische und Text-Holzbläserdiagramme

In vielen Fällen können die nicht in der mittleren Reihe befindlichen Löcher dargestellt werden, indem man die Lochbezeichnung oder graphische Zeichen benutzt.

\relative c' { 
  \textLengthOn 
  c1\markup 
  \woodwind-diagram 
  #'piccolo 
  #'((cc . (one three)) 
  (lh . (gis)) 
  (rh . (ees))) 

  c\markup 
  \override #'(graphical . #f) { 
    \woodwind-diagram 
    #'piccolo 
    #'((cc . (one three)) 
    (lh . (gis)) 
    (rh . (ees))) 
  }
}
Größe von Holzbläserdiagrammen ändern

Die Größe und Dicke der Holzbläserdiagramme kann geändert werden.

\relative c' {
\textLengthOn
\c1\markup
\woodwind-diagram
#'piccolo
#'()
\c\markup
\override #'(size . 1.5) {
\woodwind-diagram
#'piccolo
#'()
}
\c\markup
\override #'(thickness . 0.15) {
\woodwind-diagram
#'piccolo
#'()
}
}

Liste der Löcher für Holzbläserdiagramme

Kapitel 2: Spezielle Notation

#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboé (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
#(print-keys-verbose 'alto-saxophone (current-error-port))
#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))

\score {c'1}

Siehe auch

Installierte Dateien: scm/define-woodwind-diagrams.scm,
scm/display-woodwind-diagrams.scm.


2.7 Notation von Akkorden

F C F F C F

1. Fair is the sun-shine, Fair-er the moon-light
2. Fair are the meadows, Fair-er the woodland,

F Bb F C7 F C

And all the stars in heav’n above;
Robed in the flowers of blooming spring;
Akkorde können entweder als normale Noten oder im Akkordmodus notiert werden; bei letzterer Eingabemethode können unterschiedliche europäische Akkordbezeichnungsstile eingesetzt werden. Akkordbezeichnungen und Generalbass können auch angezeigt werden.

2.7.1 Akkord-Modus

Im Akkordmodus (engl. „chord“) werden Akkorde anhand von einem Symbol der erwünschten Akkordstruktur notiert, anstatt dass die einzelnen Tonhöhen ausgeschrieben werden.

Überblick über den Akkord-Modus

Akkorde können als simultane Noten eingegeben werden, wie gezeigt in [Noten mit Akkorden], Seite 152.

Akkorde können aber auch im Akkordmodus notiert werden. Das ist ein Eingabemodus, der sich an Akkordstrukturen traditioneller europäischer Musik und nicht an bestimmten einzelnen Tonhöhen orientiert. Er bietet sich an, wenn man es gewohnt ist, Akkordsymbole zur Beschreibung von Akkorden zu benutzen. Mehr Information zu unterschiedlichen Eingabemethoden findet sich in Abschnitt 5.4.1 [Eingabe-Modi], Seite 580.

\chordmode { c1 g a g c }

Akkorde, die im Akkordmodus eingegeben werden, sind musikalische Elemente und können genauso wie Akkorde im Notenmodus transponiert werden. \chordmode ist absolut, und deshalb hat \relative keine Auswirkung auf die \chordmode-Abschnitte. Im Akkord-Modus ist jedoch die absolute Tonhöhe eine Oktave höher als im Notationsmodus.

Akkordmodus und Notenmodus können gemischt verwendet werden:

\relative {<c' e g>2 <g b d>
\chordmode { c2 f }
<c e g>2 <g' b d>
\chordmode { f2 g }
}

Siehe auch

Glossar: Abschnitt “chord” in Glossar.
Notationsreferenz: [Noten mit Akkorden], Seite 152, Abschnitt 5.4.1 [Eingabe-Modi], Seite 580.

Bekannte Probleme und Warnungen

Vordefinierte Abkürzung für Artikulationen und Ornamente können mit Noten im Akkordmodus nicht benutzt werden, siehe auch [Artikulationszeichen und Verzierungen], Seite 113.
Wenn Akkord- und Notenmodus in linearer Musik abwechseln eingesetzt werden und der Akkordmodus am Anfang steht, erstellt der Notenmodus ein neues Notensystem:

\begin{verbatim}
\chordmode { c2 f }
\<c e g>2 \<g' b d>
\end{verbatim}

Um dieses Verhalten zu verhindern, muss der \texttt{Staff}-Kontext explizit aufgerufen werden:

\begin{verbatim}
\new Staff {
\chordmode { c2 f }
\<c e g>2 \<g' b d>
}
\end{verbatim}

\textbf{Übliche Akkorde}

Ein Dreiklang wird mit seinem Grundton mit einer möglichen Dauer dahinter notiert:

\begin{verbatim}
\chordmode { c2 f4 g }
\end{verbatim}

Moll- übermäßige und verminderte Dreiklänge werden notiert, indem : und ein Modifikator hinter der Dauer angegeben wird:

\begin{verbatim}
\chordmode { c2:m f4:aug g:dim }
\end{verbatim}

Septakkorde können erstellt werden:

\begin{verbatim}
\chordmode { c1:7 c:m7 c:maj7 c:dim7 c:aug7 }
\end{verbatim}

Kapitel 2: Spezielle Notation

Kein Standard: erzeugt einen Durdreiklang.

m, m7 Mollakkord: Dieser Modifikator erniedrigt die dritte Stufe.

dim, dim7 Verminderter Akkord: Dieser Modifikator erniedrigt die dritte, fünfte und (wenn vorhanden) die siebte Stufe.

aug Übermäßiger Akkord: Dieser Modifikator erhöht die fünfte Stufe.

maj, maj7 Großer Septakkord: Dieser Modifikator fügt eine erhöhte siebe Stufe hinzu. 7 nach dem maj ist optional. NICHT benutzen, um einen Durdreiklang zu notieren.

Siehe auch
Notationsreferenz: Abschnitt A.2 [Übliche Akkord-Variablen], Seite 609, [Erweiterte und modifizierte Akkorde], Seite 390.

Bekannte Probleme und Warnungen
Nur ein Qualitätsmodifikator sollte pro Akkord benutzt werden, meistens für die höchste Stufe des Akkordes. Akkorde mit weiteren Qualitätsmodifikatoren werden ohne Warnung oder Fehlermeldung gelesen, aber das Ergebnis ist nicht vorhersagbar. Akkorde, die nicht mit einem einzigen Qualitätsmodifikator erreicht werden können, sollten mit einzelnen Tonhöhen alteriert werden, wie beschrieben in [Erweiterte und modifizierte Akkorde], Seite 390.

Erweiterte und modifizierte Akkorde


\chordmode {
Sowohl c:5 als auch c erzeugen einen D-Dur-Dreiklang.

Da eine unveränderte 11 nicht gut klingt, wenn sie mit einer unveränderten 13 zusammenklingt, wird die 11 von einem \texttt{\chordmode{c1:13 c:13.11 c:m13}}-Akkord entfernt (es sei denn sie wird explizit verlangt).

Kompliziertere Akkorde können auch konstruiert werden, indem einzelne Intervalle zu dem Grundton addiert werden. Diese Additionen werden nach dem Bereich notiert und mit Punkten voneinander getrennt. Die normale Septime, die zu einem Akkord hinzugefügt wird, ist die kleine Septime, nicht die große.

Hinzugefügte Stufen können beliebig groß sein:
Einzelne Stufen können mit - oder + vergrößert oder verkleinert werden. Um eine Stufe zu verändern, die automatisch in den Akkord aufgenommen wurde, kann sie in veränderter Form nach dem Bereich hinzugefügt werden.

```
\chordmode {
  c1:7+ 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. Die einzelnen zu entfernenden Töne werden mit Punkten getrennt.

```
\chordmode {
  c1^3 c:7^5 c:9^3 c:9^3.5 c:13.11^3.7
}
```


```
\chordmode {
  c1:5 c:sus2 c:sus4 c:5.4
}
```

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

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

Eine Bassnote, die zum Akkord hinzugehört, 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/+g
}
```
Akkordmodifikatoren, die benutzt werden können, um eine große Anzahl an Standardakkorden zu erzeugen, werden gezeigt in Abschnitt A.2 [Übliche Akkord-Variablen], Seite 609.

Siehe auch
Notationsreferenz: Abschnitt A.2 [Übliche Akkord-Variablen], Seite 609.

Bekannte Probleme und Warnungen
Jede Stufe kann nur einmal in einem Akkord vorkommen. Im folgenden Beispiel wird ein erweiterter Akkord erstellt, weil 5+ zuletzt gelesen wird.
\chordmode { c1:3.5.5-.5+ }  

Nur die zweite Umkehrung kann erstellt werden, indem eine Bassnote hinzugefügt wird. Die erste Umkehrung erfordert, dass der Grundton des Akkordes geändert wird.
\chordmode {  
c'1: \c'\g e:6-3^-5 e:m6^-5  
}

2.7.2 Akkorde anzeigen
Akkorde können zusätzlich zur üblichen Notation als Töne auf einem Notensystem auch mit einem Akkordsymbol gesetzt werden.

Akkordbezeichnungen drucken
Akkordsymbole anstelle der Noten werde im ChordNames-Kontext notiert.
\new ChordNames {  
  \chordmode {  
    c2 f:4. g8  
  }  
}

C F G

Die Akkorde können entweder als simultane Noten oder unter Einsatz des Akkordmodus (chordmode) notiert werden. Der angezeigte Akkord ist der gleiche, es sei denn, Umkehrungen oder zusätzliche Basstöne werden notiert:
\new ChordNames {  
  <c e g>2 <f bes c>  
  <f c' e g>1  
  \chordmode {  
    c2 f:sus4 c1:/f
  }
}
Pausen, die in einem ChordNames-Kontext notiert werden, werden mit der noChordSymbol-Beschriftung darstellt.

\begin{verbatim}
<< \new ChordNames \chordmode {
  c1
  r1
  g1
  c1
}
\chordmode {
  c1
  r1
  g1
  c1
}
>>
\end{verbatim}

\begin{verbatim}
\chords { ... }
\end{verbatim} ist eine Kurznotation für die Bezeichnung \new ChordNames \chordmode { ... }.

\begin{verbatim}
\chords {
  c2 f4.:m g8:maj7
}
\end{verbatim}

\begin{verbatim}
C Fm G^\triangle
\new ChordNames {
  \chordmode {
    c2 f4.:m g8:maj7
  }
}
\end{verbatim}

\begin{verbatim}
C Fm G^\triangle
\end{verbatim}
Ausgewählte Schnipsel

Akkordsymbole bei Wechsel anzeigen

Akkordsymbole können so eingestellt werden, dass sie nur zu Beginn der Zeile und bei Akkordwechseln angezeigt werden.

```latex
harmonies = \chordmode {
c1:m c:m \break c:m c:m d
}
```

<<
```
\new ChordNames {
\set chordChanges = ##t
\harmonies
}
\new Staff {
\relative c' { \harmonies }
}
```

>>

Ein einfaches Liedblatt

Ein Liedblatt besteht aus Akkordbezeichnungen, einer Melodie und dem Liedtext:

```latex
<<
\chords { c2 g:sus4 f e }
\relative c' {
a4 e c8 e r4
b2 c4( d)
}
\addlyrics { One day this shall be free __ }
```

>-

Siehe auch

Glossar: Abschnitt “chord” in Glossar.
Notationsreferenz: [Musik parallel notieren], Seite 171.
Referenz der Interna: Abschnitt “ChordNames” in Referenz der Interna, Abschnitt “ChordName” in Referenz der Interna, Abschnitt “Chord_name_engraver” in Referenz der Interna,

**Bekannte Probleme und Warnungen**

Akkorde, die Umkehrungen oder zusätzliche Basstöne beinhalten, werden nicht richtig bezeichnet, wenn sie im Notenmodus notiert werden.

**Akkordbezeichnungen anpassen**

Es gibt kein allein gültiges System zur Benennung von Akkorden. Unterschiedliche Musiktraditionen benutzen unterschiedliche Bezeichnungen für die gleichen Akkorde. Es gibt zusätzlich auch unterschiedliche Symbole, die für den gleichen Akkord angezeigt werden können. Die Bezeichnungen und dargestellten Symbole können angepasst werden.


Zusätzlich zu den unterschiedlichen Bezeichnungssystemen werden unterschiedliche Notenbezeichnungen für die Grundtöne. Die vordefinierten Befehle \germanChords, \semiGermanChords, \italianChords und \frenchChords setzen diese Variablen. Die Auswirkungen werden im nächsten Beispiel gezeigt.

```latex
\begin{verbatim}
default E/D Cm B/B B#B# Bb/Bb
german E/d Cm H/h H#his B/b
semi-german E/d Cm H/h H#his Bb/Bb
italian Mi/Re Do m Si/Si Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#/Si# Si#
```

Deutsche Liederbücher zeigen Mollakkorde oft durch die Verwendung von Kleinbuchstaben an, ohne die Endung m. Dieses Verhalten kann erreicht werden, indem man die chordNameLowercaseMinor-Eigenschaft setzt:

```latex
\begin{verbatim}
\chords {
  \set chordNameLowercaseMinor = ##t
  c2 d:m e:m f
}
```

C d e F

Wenn keine der definierten Einstellungen zum gewünschten Ergebnis führt, kann die Anzeige des Akkordsymbols durch die folgenden Eigenschaften verändert werden:

**chordRootNamer**

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

additionalPitchPrefix
Wenn die Akkordbezeichnung zusätzliche Tonhöhen enthält, können sie optional mit einer Textvorsilbe versehen werden. Als Standard ist keine Vorsilbe eingestellt, sodass die visuelle Einfachheit gewahrt bleibt, aber für eine kleine Anzahl zusätzlicher Töne kann diese Option effektiv sein.

\new ChordNames {
  \set additionalPitchPrefix = "add"
  <c e g d'> % add9
}\Cadd9

chordNoteNamer
Wenn das Akkordsymbol zusätzliche Tonhöhen enthält, die nicht den Grundton darstellen (etwa eine zusätzliche Bassnote), wird diese Funktion eingesetzt, um die zusätzliche Tonhöhe auszugeben. In den Standardeinstellungen wird die Tonhöhe mit der chordRootNamer-Funktion gesetzt. Die chordNoteNamer-Eigenschaft hingegen kann dieses Verhalten verändern und etwa den Basston etwa als Kleinbuchstaben darstellen.

chordNameSeparator

\chords {
  c4:7.9- c:7.9-\slash/g
  \set chordNameSeparator = \markup { \/ } \break
  c4:7.9- c:7.9-\slash/g
}\C7\slash/G

\chords {
  c4:7.9- c:7.9-\slash/g
}\C7\slash/G

slashChordSeparator
Akkorde können auch aufbauend auf einer anderen Bassnote als dem üblichen Grundton gespielt werden. Sie werden Umkehrungen genannt, sind aber auch als Slash-Akkorde bekannt, weil ihre übliche Notation aus dem Akkordsymbol, einem Schrägstrich und dem Basston besteht. Deshalb ist der Standardwert von slashChordSeparator ein Schrägstrich, aber jedes andere Beschriftungszeichen kann auch eingestellt werden.

\chords {
  c4:7.9- c:7.9-\slash/g
}
Kapitel 2: Spezielle Notation

\set slashChordSeparator = \markup { " over " }
\break
c4:7.9- c:7.9-/g
}

\markup \text{C$^7\flat$ over G}
\markup \text{C$^7\flat$ over G}

\chordNameExceptions
Diese Funktion ist eine Liste mit Paaren. Das erste Objekt eines Paares ist eine Anzahl von Tonhöhen, die die Stufen eines Akkordes definieren. Das zweite Objekt ist eine Beschriftung, die nach chordRootNamer ausgegeben wird, um das Akkordsymbol zu erstellen.

\minorChordModifier
Moll-Akkorde werden oft durch ein nachgestelltes „m“ rechts des Akkordgrundtons angezeigt. Manche bevorzugen aber andere Zeichen, wie etwa ein Minus-Zeichen.

\chords {
c4:min f:min7
\set minorChordModifier = \markup { "-" }
\break
c4:min f:min7
}

\chordPrefixSpacer

Vordefinierte Befehle
\whiteTriangleMarkup, \blackTriangleMarkup, \germanChords, \semiGermanChords, \italianChords, \frenchChords.

Ausgewählte Schnipsel
Akkordsymbolausnahmen
Die Eigenschaft chordNameExceptions kann benutzt werden, um eine Liste an besonderen Notationen für bestimmte Akkorde zu speichern.
% 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)" }
}

Ausgewählte Schnipsel
Akkordsymbolausnahmen
Die Eigenschaft chordNameExceptions kann benutzt werden, um eine Liste an besonderen Notationen für bestimmte Akkorde zu speichern.
% 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
>>

Akkordbezeichnung maj7
Das Aussehen des großen Septakkords kann mit majorSevenSymbol verändert werden.
\chords {
c:7+
\set majorSevenSymbol = \markup { j7 }
c:7+
}

C\(^\Delta\) C\(^7\)

Taktstriche in einen ChordNames-Kontext hinzufügen
Um Taktstriche in einem ChordNames-Kontext anzeigen zu lassen, muss der Bar_engraver hinzugefügt werden.
\new ChordNames \with {
    \override BarLine.bar-extent = #'(-2 . 2)
    \consists "Bar_engraver"
}

\chordmode {
f1:maj7 f:7 bes:7
}

F\(^\Delta\) | F\(^7\) | B\(^b\)\(^7\) |

Wiederholungs-(Volta-)Klammern unterhalb der Akkordsymbole
Indem man den Volta_ engraver zu dem entsprechenden Notensystem hinzufügt, können Wiederholungsklammern unterhalb der Akkorde gesetzt werden.

\score {
  <<
    \chords {
      c1
      c1
    }
    \new Staff \with {
      \consists "Volta_ engraver"
    }
    \repeat volta 2 { c'1 }
    \alternative { c' }
  }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_ engraver"
    }
  }
}

Abbildung 1.2: C C

Akkordsymboltrenner verändern

Der Trenner zwischen unterschiedlichen Teilen eines Akkordsymbols kann beliebiger Text sein.

\chords {
  c:7sus4
  \set chordNameSeparator = \markup { \typewriter | }
  c:7sus4
}

\begin{tabular}{l}
C7 sus4 C7|sus4
\end{tabular}

Siehe auch

Notationsreferenz: Abschnitt A.1 [Liste der Akkordbezeichnungen], Seite 608, Abschnitt A.2 [Übliche Akkord-Variablen], Seite 609.


Bekannte Probleme und Warnungen

Akkordsymbole werden von den Tonhöhenbezeichnungen innerhalb des Akkordes und der Information über die Akkordstruktur, die innerhalb von \chordmode notiert wurde, bestimmt. Wenn der direkte Notenmodus benutzt wird, stammen unerwünschte Bezeichnungen daher, dass Umkehrungen und zusätzliche Bassnoten nicht richtig interpretiert werden.

```
myChords = \relative c' {
  \chordmode { c1 c/g c/f }
  <c e g>1 <g c e> <f c' e g>
}
<<
  \new ChordNames { \myChords }
  \new Staff { \myChords }
>>
```

2.7.3 Generalbass

```
\Adagio.
```

```
\Violino I.\ 
```

```
\Violino II.\ 
```

```
\Violone, e Cembalo.
```

```
\Violine I.\ 
```

```
\Violine II.\ 
```

```
\Violone, e Cembalo.
```
Generalbassnotation kann dargestellt werden.

**Grundlagen des Bezifferten Basses**

LilyPond stellt Unterstützung für Generalbassnotation, auch als Basso Continuo bezeichnet, 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 für Generalbass 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 `FiguredBass`, der dafür sorgt, dass die entsprechenden `BassFigure`-Objekte auch erstellt werden. Generalbass kann auch in einem `Staff`-Kontext dargestellt werden.

`\figures{ ... }` ist eine Kurznotation für `\new FiguredBass { \figuremode { ... } }`.

Auch wenn die Unterstützung für Generalbass auf den ersten Blick wie die Akkordunterstützung ausschauen mag, ist sie sehr viel einfacher. `\figuremode` speichert einfach die Zahlen und der `FiguredBass`-Kontext setzt sie in der Form, wie sie notiert wurden. Sie werden nicht in Tonhöhen umgewandelt.

**Siehe auch**

Glossar: Abschnitt “figured bass” in Glossar.

**Eingabe des Generalbass’**

`\figuremode` (Zahlenmodus) wird benutzt, um den Eingabemodus auf den Zahlenmodus umzustellen. Mehr Information zu unterschiedlichen Eingabemodi findet sich in Abschnitt 5.4.1 [Eingabe-Modi], Seite 580.

\new FiguredBass {
  \figuremode {
    <6 4>2
  }
}

6
\frac{4}{3}

Versetzungszeichen (inklusive Auflösungszeichen) können hinzugefügt werden:

\figures {
  <7! 6+ 4--> <5+++> <3--> 
}

\frac{7}{5} \times 5 \frac{3}{4}

Übermäßige und verminderte Stufen können dargestellt werden:

\figures {
  <6\+ 5/> <7/> 
}

\frac{6}{5} \div 7

Ein Schrägstrich von links nach rechts (üblicherweise für erhöhte Sexten benutzt) kann erstellt werden:

\figures {
  <6> <6\> 
}

6
\frac{6}{6}

Vertikaler Platz und Klammern können zu den Zahlen hinzugefügt werden:

\figures {
  <[12 _!] 8 [6 4]> 
}

\frac{12}{8} \frac{8}{4}

Beliebiger Text kann als Zahl notiert werden:

\figures {
  \markup { \tiny \number 6 \super (1) } 5>
}

6^{(1)}
Es ist auch möglich, Fortsetzungslinien für wiederholte Ziffern zu benutzen.

\begin{verbatim}
{\clef bass
e4 d c b,
e4 d c b,
}
\figures{
\bassFigureExtendersOn
<6 4>4 <6 3> <7 3> <7 3>
\bassFigureExtendersOff
<6 4>4 <6 3> <7 3> <7 3>
}
\end{verbatim}

In diesem Fall werden wiederholte Ziffern immer durch eine Linie ersetzt, es sei denn, die Linie wird explizit beendet.

\begin{verbatim}
\figures{
\bassFigureExtendersOn
<6 4>4 <6 4> <6! 4!> <6 4>
}
{
\clef bass
d4 d c c
}
\end{verbatim}

Die folgende Tabelle zeigt die vorhandenen Zahlenmodifikatoren:

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Purpose</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>+, -, !</td>
<td>Accidentals</td>
<td>7 5 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>+, /</td>
<td>Augmented and diminished steps</td>
<td>6 7</td>
</tr>
</tbody>
</table>
Kapitel 2: Spezielle Notation

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{example.png}
\caption{Example of special notation}
\end{figure}

\begin{Verbatim}
Vordefinierte Befehle
\bassFigureExtendersOn, \bassFigureExtendersOff.
\end{Verbatim}

\begin{Verbatim}
\textbf{Ausgewählte Schnipsel}

Positionen von Generalbass-Alterationszeichen verändern

Versetzungszeichen und Pluszeichen können vor oder nach den Ziffern erscheinen, je nach den Einstellungen der \texttt{figuredBassAlterationDirection} und \texttt{figuredBassPlusDirection}-Eigenschaften.

\begin{verbatim}
\figures {
  \set figuredBassAlterationDirection = #RIGHT
  \set figuredBassPlusDirection = #RIGHT
  \set figuredBassAlterationDirection = #LEFT
  \set figuredBassPlusDirection = #LEFT
}
\end{verbatim}

\begin{verbatim}
+6 t5 6 \text{\fourth}
+6 5t 6 \text{\fourth}
+6 5\# 6 \text{\fourth}
+6 5\# 6 \text{\fourth}
\end{verbatim}

\begin{Verbatim}
Siehe auch


Referenz der Interna: Abschnitt “BassFigure” in Referenz der Interna, Abschnitt “BassFigureAlignment” in Referenz der Interna, Abschnitt “BassFigureLine” in Referenz der Interna, Abschnitt “BassFigureBracket” in Referenz der Interna, Abschnitt “BassFigureContinuation” in Referenz der Interna, Abschnitt “FiguredBass” in Referenz der Interna.
\end{Verbatim}

\begin{Verbatim}
\textbf{Generalbass anzeigen}

Generalbass kann mit dem \texttt{FiguredBass}-Kontext, aber auch in den meisten anderen Staff-Kontexten dargestellt werden.

Wenn die Ziffern im \texttt{FiguredBass}-Kontext dargestellt werden, ist die vertikale Position der Ziffern unabhängig von den Noten des parallelen Systems.

\begin{verbatim}
<<
\relative {
  c'\quarter 4 c'\quarter 8 r8 c,4 c'
}
\new FiguredBass {
  \figuremode {
    <4>4 <10>6 >8 s8
\end{verbatim}

\begin{verbatim}
In diesem Beispiel muss der FiguredBass-Kontext explizit erstellt werden, damit kein zusätzliches (leeres) Notensystem erstellt wird.

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

Wenn Generalbass zu einem vorhandenen System hinzugefügt wird, ist es möglich, die Ziffern über oder unter dem System anzuzeigen:
Bekannte Probleme und Warnungen

Um sicherzugehen, dass die Fortsetzungslinien funktionieren, sollte der gleiche Rhythmus für die Bassfiguren und die eigentlichen Noten der Bassstimme benutzt werden.

Wenn Fortsetzungslinien eingesetzt werden, können aufeinander folgende Bezifferungen mit der selben Zahl in einer anderen Position dazu führen, dass sich die Reihenfolge der Zahlen umkehrt.
Um dieses Problem zu umgehen, kann die Fortsetzungslinie nach der Bezifferung, mit der die Linie beginnen soll, angeschaltet und am Ende der Linie wieder ausgeschaltet werden.

\begin{example}
{ fis4 g g, e' }
\figures {
   \bassFigureExtendersOn
   <6 5>4 <5\! 4> <5 _!> <6>
}
\end{example}

2.8 Zeitgenössische Musik


2.8.1 Tonhöhe und Harmonie in zeitgenössischer Musik

Dieser Abschnitt zeigt Lösungen zur Notation von zeitgenössischen Tonhöhen und Harmonien.

Verweise zu Tonhöhe und Harmonie in zeitgenössischer Musik

- Normale Vierteltonmusik wird behandelt in [Notenbezeichnungen in anderen Sprachen], Seite 7.
- Nicht-Standardvorzeichen werden behandelt in [Tonartbezeichnung], Seite 20.
- Contemporary practises in displaying accidentals are addressed in [Automatische Versetzungssymbole], Seite 24.

Mikrotonale Notation
Zeitgenössische Tonartvorzeichnung und Harmonie

2.8.2 Zeitgenössische Notation von Rhythmen
Dieser Abschnitt erklärt Besonderheiten, die wichtig für die Notation von Rhythmus in zeitgenössischer Musik sind.

Verweise für zeitgenössische Benutzung von Rhythmus
- Zusammengesetzte Taktarten werden erklärt in #\{Taktangabe\}, Seite 60.
- Grundlegende polymetrische Notation ist erklärt in \{Polymetrische Notation\}, Seite 71.
- Gespreizte Balken sind erklärt in \{Gespreizte Balken\}, Seite 91.
- Mensurstrich-Taktstriche (zwischen den Systemen) finden sich erklärt in \{undefined\} \{Grouping staves\}, Seite \{undefined\}.

N-tolen in zeitgenössischer Musik

Zeitgenössische Taktarten

Erweiterte polymetrische Notation

Balken in zeitgenössischer Musik

Taktstriche in zeitgenössischer Musik

2.8.3 Graphische Notation

2.8.4 Zeitgenössische Partiturtechniken

2.8.5 Neue Instrumententechniken

2.8.6 Leseliste und interessante Referenzpartituren
Dieser Abschnitt weist auf einige Bücher, Musikbeispiele und andere Resourcen hin, die relevant für die Notation zeitgenössischer Musik sind.

Bücher und Artikel über zeitgenössische Notation
- *Instrumentation and Orchestration* von Alfred Blatter [Schirmer, 2nd ed. 1997]

Partituren und Musikbeispiele

2.9 Notation von alter Musik

Salve, Regina, mater misericordiae: Ad te clamamus, exules, filii Hevae. Ad te superna cautio.
Unterstützung für Notation der Alten Musik enthält einige Eigenheiten der Mensuralnotation, der Notation des gregorianischen Chorals und der Kiever Quadratnotation. Diese Eigenheiten können eingestellt werden, indem man Stileigenschaften von graphischen Objekten wie Notenköpfen und Pausen verändert, oder indem man vordefinierte fertige Kontexte für mensurale oder Choralnotation einsetzt.

Viele graphische Objekte, wie Notenköpfe, Fähnchen, Versetzungszeichen, Taktarten und Pausen haben eine style-Eigenschaft, die verändert werden kann, um verschiedene Stile Alter Notation nachzunehmen. Siehe auch:

- [Mensurale Notenköpfe], Seite 416,
- [Mensurale Versetzungszeichen und Tonartbezeichnung], Seite 419,
- [Mensurale Pausen], Seite 418,
- [Mensurale Schlüssel], Seite 414,
- [Gregorianische Schlüssel], Seite 422,
- [Mensurale Fähnchen], Seite 417,
- [Mensurale Taktartenbezeichnungen], Seite 415.

Ein paar notationelle Konzepte sind insbesondere für die Notation Alter Musik eingeführt worden:

- [Custodes], Seite 413,
- [Divisiones], Seite 423,
- [Ligaturen], Seite 412.
Siehe auch


Notationsreferenz: [Mensurale Notenköpfe], Seite 416, [Mensurale Versetzungszeichen und Tonartbezeichnung], Seite 419, [Mensurale Pausen], Seite 418, [Mensurale Schlüssel], Seite 414, [Mensurale Fähnchen], Seite 417, [Mensurale Taktartenbezeichnungen], Seite 415, [Gregorianische Schlüssel], Seite 422, [Custodes], Seite 413, [Divisiones], Seite 423, [Ligaturen], Seite 412.

2.9.1 Überblick über die unterstützten Stile

Drei Stile sind vorhanden, um den gregorianischen Choral zu setzen:


- Der **Editio Medicea**-Stil stellt bestimmte Eigenschaften zur Verfügung, die in den Medicea (oder Ratisbona)-Editionen benutzt wurden. Dieser Stil war vor den Solsemes-Editionen in Benutzung. Der größte Unterschied von dem Vaticana-Stil sind die Schlüssel, die nach unten gerichtete Striche haben, und die Notenköpfe, die hier quadratisch und ebenmäßig geformt sind.

- Der **Hufnagel- oder gotische** Stil ahmt den Stil der Schreiber bestimmter Manuskripte aus dem Deutschland und Mitteleuropa des Mittelalters nach. Er ist nach der Form der wichtigsten Note (der Virga) benannt, die wie ein kleiner Nagel aussieht.

Drei Stile ahmen die Erscheinung von Renaissancehandschriften und -drucken der Mensuralmusik nach:

- Der **Mensural**-Stil versucht, den Stil von Handschriften nachzunehmen und hat recht kleine, rhombenförmige Notenköpfe und wie handgeschriebene Pausenzeichen.

- Der **Neomensural**-Stil ist eine modernisierte und stilisierte Version des erstden: Die Notenköpfe sind etwas breiter und die Pausen bestehen aus graden Linien. Dieser Stil ist besonders gut geeignet, um moderne Editionen der Mensuralmusik mit einem Incipit zu versehen.

- Der **Petrucci**-Stil ist nach Ottaviano Petrucci (1466-1539) benannt, dem ersten Drucker, der bewegliche Stempel benutzt hat, um musikalische Notation zu drucken (in seinem Buch Harmonice musices odhecaton, 1501). Dieser Stil setzt größere Notenköpfe ein als die anderen mensuralen Stile.

**Baroque** (Barockstil) und **Classical** (klassischer Stil) sind keine vollständigen Stile, sondern unterscheiden sich vom Standard nur in einigen Details: der Barockstil verändert bestimmte Notenköpfe, der klassische Stil die Form der Viertelpause.

Nur der Mensuralstil hat für alle Aspekte der Notation eine alternative Form. Die anderen Stile sind nur teilweise ausgeführt: die gregorianischen Stile haben keine Pausen oder Fähnchen, weil diese Zeichen im Choral nicht vorkommen, und der Petrucci-Stil hat keine eigenen Fähnchen und Versetzungszeichen.

Jedes Notationselement kann unabhängig von den anderen verändert werden, sodass man gut mensurale Fähnchen, Petrucci-Notenköpfe, klassische Pausen und Vatikana-Schlüssel nebeneinander benutzen kann, wenn das gewünscht ist.

Siehe auch

2.9.2 Alte Notation – Allgemeines

Vordefinierte Umgebungen


Siehe auch

- [Gregorianische Gesangs-Kontexte], Seite 421,
- [Mensural-Kontexte], Seite 414.

Siehe auch

Glossar: Abschnitt “mensural notation” in Glossar.


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.


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


Siehe auch

Glossar: Abschnitt “ligature” in Glossar.

Notationsreferenz: [Weiße Mensuralligaturen], Seite 420, [Ligaturen der gregorianischen Quadratnotation], Seite 425.
**Bekannte Probleme und Warnungen**

Ligaturen benötigen eine Platzauflösung, die sich von der klassischen Notation deutlich unterscheidet. Das ist bisher sehr schlecht verwirklicht, sodass fast immer zu viel Platz zwischen Ligaturen ist und Zeilenumbrüche unbefriedigend ausfallen. Text lässt sich auch nicht richtig an Ligaturen ausrichten.

Versetzungszeichen 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 (\[ musik. Ausdr. \]). Aus Gründen der Konsistenz soll dies geändert werden in den Postfix-Stil (Note\[ ... Note\]).

**Custodes**

Ein *Custos* (Plural: *Custodes*; Lateinisch: „Weiser“) 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.


Damit Custodes angezeigt werden, muss ein *Custos_engraver* im Staff-Kontext gefordert werden. Der Aufruf folgt im Rahmen des Layout-Kontextes, wie das folgende Beispiel zeigt. Der Stil des Custos wird mit dem *override*-Befehl eingestellt, wie in dem folgenden Beispiel gezeigt:

```
\override \staff \staffname {\custodes.mensural.u1 \times 1 \noteheads.s0 \clefs.G}
```


```
vaticana medicaea hufnagel mensural
```

Siehe auch

- Referenz der Interna: Abschnitt “Custos” in Referenz der Interna.

**Unterstützung für Generalbass**

Es gibt beschränkte Unterstützung für Generalbassziffern aus der Barockzeit. Siehe hierzu Abschnitt 2.7.3 [Generalbass], Seite 401.

Siehe auch

- Glossar: Abschnitt “figured bass” in Glossar.
- Notationsreferenz: Abschnitt 2.7.3 [Generalbass], Seite 401.
2.9.3 Mensurale Musik setzen

Mensural-Kontexte

Die vordefinierten Kontexte \texttt{MensuralVoice} und \texttt{MensuralStaff} können eingesetzt werden, um ein Stück in Mensuralnotation zu schreiben. Die Kontexte initialisieren alle relevanten Eigenschaften und graphischen Objekte, so dass unmittelbar mit der Notation begonnen werden kann. Siehe das folgende Beispiel:

\begin{verbatim}
\score {
  \new \score { \new \MensuralVoice \transpose c \c' { \hide \Score.BarNumber { \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 }
} \end{verbatim}

Siehe auch

Glossar: Abschnitt “mensural notation” in Glossar.

Mensurale Schlüssel

In der Tabelle unten werden alle Mensuralschlüssel gezeigt, die mit dem \texttt{\clef}-Befehl erreicht werden. Manche Schlüssel benutzen dasselbe Zeichen, unterscheiden sich aber in der Notenlinie, auf der der Schlüssel notiert wird. In diesem Fällen ist eine Nummer im Schlüsselnamen eingefügt, nummeriert von unten nach oben. Man kann aber trotzdem eine beliebige Nummer erzwingen, wie es im Abschnitt [Notenschlüssel], Seite 16, beschrieben wird. Die Note, die rechts von jedem Schlüssel gesetzt ist, zeigt das \c' in Bezug zu dem jeweiligen Schlüssel.
Petrucci hat C-Schlüssel benutzt, die unterschiedlich ausbalancierte vertikale Balken auf der linken Seite hatten, je nachdem, auf welcher Notenlinie er sich befand.

<table>
<thead>
<tr>
<th>Beschreibung</th>
<th>Unterstützte Schlüssel</th>
<th>Beispiel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mensuraler C-Schlüssel im historischen Stil</td>
<td>mensural-c1, mensural-c2, mensural-c3, mensural-c4</td>
<td></td>
</tr>
<tr>
<td>Mensuraler F-Schlüssel im historischen Stil</td>
<td>mensural-f</td>
<td></td>
</tr>
<tr>
<td>Mensuraler G-Schlüssel im historischen Stil</td>
<td>mensural-g</td>
<td></td>
</tr>
<tr>
<td>Mensuraler C-Schlüssel im modernen Stil</td>
<td>neomensural-c1, neomensural-c2, neomensural-c3, neomensural-c4</td>
<td></td>
</tr>
<tr>
<td>Mensuraler C-Schlüssel im Petrucci-Stil, zur Benutzung auf verschiedenen Notenlinien (im Beispiel den Schlüssel auf der zweiten Linie)</td>
<td>petrucci-c1, petrucci-c2, petrucci-c3, petrucci-c4, petrucci-c5</td>
<td></td>
</tr>
<tr>
<td>Mensuraler F-Schlüssel im Petrucci-Stil, kann auf verschiedenen Notenlinien benutzt werden (im Beispiel auf der dritten Linie)</td>
<td>petrucci-f3, petrucci-f4, petrucci-f5</td>
<td></td>
</tr>
<tr>
<td>Mensuraler G-Schlüssel im Petrucci-Stil</td>
<td>petrucci-g</td>
<td></td>
</tr>
</tbody>
</table>

Siehe auch
Glossar: Abschnitt "mensural notation" in Glossar, Abschnitt "clef" in Glossar.
Notationsreferenz: [Notenschlüssel], Seite 16.

Bekannte Probleme und Warnungen
Der mensurale G-Schlüssel ist als Petrucci-G-Schlüssel deklariert.

Mensurale Taktartenbezeichnungen
LilyPond besitzt beschränkte Unterstützung für Mensurzeichen (die den heutigen Taktarten ähneln, aber doch einige Eigenheiten haben). 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.
Mit der `style`-Eigenschaft des Objektes `TimeSignature` können die Taktarten angewählt werden. Unterstützte Stile sind: `neomensural` und `mensural`. In der Tabelle oben wurde der neomensurale Stil verwendet. Im folgenden Beispiel sind die unterschiedlichen Stile dargestellt.

**Siehe auch**

Glossary: Abschnitt “mensural notation” in *Glossar*.
Notationsreferenz: [Taktangabe], Seite 60.

**Bekannte Probleme und Warnungen**

Die Verhältnisse der Notenwerte können nicht bei Mensurwechsel geändert werden, weil sie nicht konstant sind. Zum Beispiel kann das Verhältnis 1 brevis = 3 semibrevis (tempus perfectum) manuell erstellt werden, indem folgende Variable erstellt wird:

```
breveTP = #(ly:make-duration -1 0 3/2)
```

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

Die Symbole `mensural68alt` und `neomensural68alt` (alternative Symbole für 6/8) können nicht mit dem `\time`-Befehl. Anstelle dess muss `\markup {\musicglyph "timesig.mensural68alt" }` benutzt werden.

**Mensurale Notenköpfe**

Für die Mensuralnotation kann ein Notenkopfstil ausgewählt werden, der sich vom Standard (`default`) unterscheidet. Dies wird erreicht, indem die `style`-Eigenschaft des Notenkopf-(`NoteHead`)-Objekts auf einen der Werte `baroque`, `neomensural`, `mensural`, `petrucci`, `blackpetrucci` oder `semipetrucci` gesetzt wird.

Der barocke (`baroque`) Stil unterscheidet sich vom Standard (`default`) folgendermaßen:
- Er stellt einen *maxima*-Notenkopf zur Verfügung und
- setzt eine eckige Form für die *Brevis* (`\breve`) ein.

Die Stile `neomensural`, `mensural` und `petrucci` unterscheiden sich vom barocken Stil folgendermaßen:
- Für Semibrevis und kleinere Notenwerte werden rhombenförmige Notenköpfe eingesetzt und
Die Hälse werden über den Kopf zentriert.

Der blackpetrucci-Stil erstellt Notenköpfe zur Benutzung für die schwarze Mensuralnotation oder coloratio-Abschnitten in der weißen Mensuralnotation. Weil der Notenkopfstil nicht die Anzahl der Fähnchen beeinflusst, muss eine Semiminia in diesem Stil als a8*2 notiert werden, nicht als a4, weil sie sonst wie eine Minima aussehen würde. Der Faktor, mit dem der Notenwert multipliziert wird, kann sich ändern, wenn coloratio etwa zur Notation von Triolen eingesetzt wurde.

Mit dem semipetrucci-Stil können halb-schwarze Notenköpfe notiert werden (Brevis, Longa und Maxima).

Das folgende Beispiel zeigt den Petrucci-Stil:

\set Score.skipBars = ##t
\autoBeamOff
\override NoteHead.style = #'petrucci
a'\maxima a'\longa a'\breve a'1 a'2 a'4 a'8 a'16 a'
\override NoteHead.style = #'semipetrucci
a'\breve*5/6
\override NoteHead.style = #'blackpetrucci
a'8*4/3 a'
\override NoteHead.style = #'petrucci
a'\longa

Siehe auch

Glossar: Abschnitt “mensural notation” in Glossar, Abschnitt “note head” in Glossar.
Notationsreferenz: Abschnitt A.9 [Notenkopfstile], Seite 646.

Mensurale Fähnchen

Mit der Fähnchen-(flag-style)-Eigenschaft der graphischen Objekte „Hals“ (Stem) können auch Mensuralfähnchen gesetzt werden. Neben dem Standardstil (default) ist nur (mensural) unterstützt.

\override 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 den neomensuralen oder Petrucci-Stil. Für die Notation des Gregorianischen Chorals gibt es keine Fähnchen.
Kapitel 2: Spezielle Notation

Siehe auch
Glossar: Abschnitt “mensural notation” in Glossar, Abschnitt “flag” in Glossar.

Bekannte Probleme und Warnungen
Die Positionierung der Fähnchen an den Hälsern ist leicht verschoben.

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 kompliziertere Methoden zur Ermittlung des besten Layouts verwendet. Diese Methoden sollten aber eigentlich nicht zur Notation von mensuraler Musik eingesetzt werden.

Mensurale Pausen
Besondere Pausensymbole für die Notation der Alten Musik können mit der style-Eigenschaft des graphischen Objektes (grob) „Pause“ (Rest) 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 mensurale und neomensurale Stil ahmt die Form von Pausen nach, wie man sie in Drucken bis zum 16. Jahrhundert finden kann.

Das folgende Beispiel demonstriert den mensuralen und den neomensuralen Stil:

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

Eine Liste aller Pausen findet sich in Abschnitt “Ancient notation” in Schnipsel.

Siehe auch
Notationsreferenz: [Pausen], Seite 52.
Bekannte Probleme und Warnungen

Das Zeichen für die Maxima-Pause im mensuralen Stil ist eigentlich eine perfekte Longa-Pause: zwei (oder drei) Longa-Pausen müssen benutzt werden, um eine Maxima-Pause zu setzen. Longa-Pausen werden nicht automatisch gruppiert, sodass man das manuell vornehmen muss, indem man Pausen mit Tonhöhe einsetzt.

Mensurale Versetzungszeichen und Tonartbezeichnung

Der mensural-Stil stellt ein Kreuz und ein B zur Verfügung, die sich vom Standardstil unterscheiden. Wenn das Auflösungszeichen notiert wird, wird es aus dem vaticana-Stil gesetzt.

\begin{verbatim}
\override Staff.Accidental.glyph-name-alist = #alteration-mensural-glyph-name-alist
\end{verbatim}

Siehe auch


Notationsreferenz: Abschnitt 1.1 [Tonhöhen], Seite 1, [Versetzungssprüche], Seite 5, [Automatische Versetzungszeichen], Seite 24, [Tonartbezeichnung], Seite 20.

Referenz der Interna: Abschnitt “KeySignature” in Referenz der Interna.

Vorgeschlagene Versetzungszeichen (musica ficta)

In der europäischen Notation bis etwa 1600 wurde von Sängern erwartet, dass sie eigenständig Noten nach bestimmten Regeln chromatisch veränderten. Das wird als musica ficta bezeichnet. In modernen Transkriptionen werden diese Versetzungszeichen üblicherweise über die Note notiert.

Es ist möglich, derartige Versetzungszeichen zu notieren, und die Anzeige kann zwischen normaler Satzweise und musica ficta hin- und hergewechselt werden. Hierzu muss suggestAccidentals auf wahr gesetzt werden:

\begin{verbatim}
\set suggestAccidentals = ##f
\end{verbatim}

Damit wird jedes folgende Versetzungszeichen über dem System gesetzt werden, bis die Eigenschaft mit \set suggestAccidentals = ##f wieder zum Standardverhalten verändert wurde. Eine praktischere Lösung ist es, \once \set suggestAccidentals = ##t zu benutzen, was man als Variable definieren kann:

\begin{verbatim}
ficta = { \once \set suggestAccidentals = ##t }
\score { \relative 
   \new MensuralVoice { 
      \once \set suggestAccidentals = ##t 
      bes'4 a2 g2 \ficta fis8 \ficta e! fis2 g1
   }
}
\end{verbatim}
Kapitel 2: Spezielle Notation

Siehe auch


Weiße Mensuralligaturen

Begrenzte Unterstützung für Ligaturen der weißen Mensuralnotation ist vorhanden.

Um weiße Mensuralligaturen zu benutzen, muss innerhalb des Layout-Blocks im Voice-Kontext der Mensural_ligature_engraver aktiviert werden und gleichzeitig der Ligature_bracket_engraver (der die Klammern über den Noten setzt) entfernt werden, wie im Beispiel.

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


An bestimmten Stellen können zwei aufeinanderfolgende Noten entweder als zwei Quadrate oder eine Obliqua (Flexa) dargestellt werden. In derartigen Fällen ist die Quadratform der Standard, aber die Obliqua kann verlangt werden, indem man die ligature-flexa-Eigenschaft des zweiten Notenkopfes setzt. Die Länge der Obliqua kann durch die Notenkopfeigenschaft flexa-width definiert werden.

Eine Datei kann zum Beispiel so aussehen:

\score {
  \transpose c c' {
    \set Score.timing = ##f
    \set Score.defaultBarType = "-"
    \override NoteHead.style = #'petrucci
    \override Staff.TimeSignature.style = #'mensural
    \clef "petrucci-g"
    \[ c\maxima g \]
    \[ d\longa \]
    \[ c\maxima d\longa \]
    \[ e'1 a \breve \]
  }
  \layout {
    \context {
      \Voice
      \remove "Ligature_bracket_engraver"
    }
  }
}
Wenn der Ligature_bracket_engraver nicht durch den Mensural_ligature_engraver ersetzt wird, werden die Noten wie folgt ausgegeben:

\begin{example}
\includegraphics[width=\textwidth]{example.png}
\end{example}

Siehe auch

Glossar: Abschnitt “ligature” in Glossar.
Notationverweis: [Ligaturen der gregorianischen Quadratnotation], Seite 425, [Ligaturen], Seite 412.

Bekannte Probleme und Warnungen

Die horizontale Positionierung ist sehr schlecht.
Versetzungssignale können mit vorhergehenden Noten kollidieren.

2.9.4 Gregorianischen Choral setzen

Wenn ein gregorianischer Choral notiert wird, wählt der Vaticana_ligature_engraver automatisch die richtigen Notenköpfe aus, so dass man den Notenkopfstil nicht explizit setzen muss. Der Stil kann dennoch gesetzt werden, etwa auf vaticana_punctum um punctum-Neumen zu erzeugen. Ähnlich funktioniert auch der Mensural_ligature_engraver, der Mensuralligaturen setzt.

Siehe auch

Glossar: Abschnitt “ligature” in Glossar.
Notationverweis: [Weiße Mensuralligaturen], Seite 420, [Ligaturen], Seite 412.

Gregorianische Gesangs-Kontexte

Die vordefinierten Kontexte VaticanaVoice (für eine gregorianische Stimme) und VaticanaStaff (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:

\begin{example}
\include "gregorian.ly"
\score {
<<
\new VaticanaVoice = "cantus" {
  \[ c\'melisma c' \flexa a \]
  \[ a \flexa \deminutum g\melismaEnd \]
  f \divisioMinima
  \[ f\melisma \pes a c' c' \pes d'\melismaEnd \]
\}
\end{example}
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
}
">
}

\begin{music}
\begin{lyric}
San-ctus, San-ctus,
\end{lyric}
\end{music}

**Gregorianische Schlüssel**

Die folgende Tabelle zeigt alle Schlüssel für den gregorianischen Choral, die mit dem \clef-Befehl unterstützt sind. Einige Schlüssel benutzen das selbe Zeichen, unterscheiden sich aber in der Notenlinie, auf der der Schlüssel gesetzt wird. In diesem Fall wird eine Nummer benutzt, die die Notenlinie von unten nach oben kennzeichnet. Man kann die Schlüssel aber auch manuell auf eine bestimmte Notenlinie zwingen, wie gezeigt in [Notenschlüssel], Seite 16. Die Note, die rechts von den Schlüsseln im Beispiel gezeigt wird, ist ein c' in Bezug auf den aktuellen Schlüssel.

<table>
<thead>
<tr>
<th>Beschreibung</th>
<th>unterstützter Schlüssel</th>
<th>Beispiel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do-Schlüssel der Editio Vaticana</td>
<td>vaticana-do1, vaticana-do2, vaticana-do3</td>
<td>$\frac{\text{C}}{\text{C}}$</td>
</tr>
<tr>
<td>Fa-Schlüssel der Editio Vaticana</td>
<td>vaticana-fa1, vaticana-fa2</td>
<td>$\frac{\text{G}}{\text{G}}$</td>
</tr>
<tr>
<td>Do-Schlüssel der Editio Medicaea</td>
<td>medicaea-do1, medicaea-do2, medicaea-do3</td>
<td>$\frac{\text{C}}{\text{C}}$</td>
</tr>
<tr>
<td>Fa-Schlüssel der Editio Medicaea</td>
<td>medicaea-fa1, medicaea-fa2</td>
<td>$\frac{\text{G}}{\text{G}}$</td>
</tr>
<tr>
<td>Hufnagel Do-Schlüssel für den historischen Stil</td>
<td>hufnagel-do1, hufnagel-do2, hufnagel-do3</td>
<td>$\frac{\text{C}}{\text{C}}$</td>
</tr>
</tbody>
</table>
Kapitel 2: Spezielle Notation

Hufnagel Fa-Schlüssel für den historischen Stil

\hufnagel-fa1, \hufnagel-fa2

Kombinierter Do/Fa-Schlüssel für den historischen Stil

\hufnagel-do-fa

Siehe auch

Glossar: Abschnitt “clef” in Glossar.
Notationsreferenz: [Notenschlüssel], Seite 16.

Gregorische Versetzungszeichen und Tonartbezeichnung

Es gibt Versetzungszeichen in drei unterschiedlichen Stilen für die Notation des gregorianischen Chorals:

\vaticana medicaea hufnagel

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.

Der Stil für Versetzungs- und Vorzeichen wird von der \glyph-name-alist-Eigenschaft der Grobs Accidental und KeySignature kontrolliert, beispielsweise:

\override Staff.Accidental.glyph-name-alist = #alteration-mensural-glyph-name-alist

Siehe auch

Glossar: Abschnitt “accidental” in Glossar, Abschnitt “key signature” in Glossar.
Notationsreferenz: Abschnitt 1.1 [Tonhöhen], Seite 1, [Versetzungszeichen], Seite 5, [Automatische Versetzungszeichen], Seite 24, [Tonartbezeichnung], Seite 20.
Referenz der Interna: Abschnitt “KeySignature” in Referenz der Interna.

Divisiones

Die Notation des gregorianischen Chorals benutzt keine Pausen, anstatt dessen werden Divisiones eingesetzt.


Divisiones können benutzt werden, indem die Datei gregorian.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
Kapitel 2: Spezielle Notation

minima. Darum findet sich in der Datei `gregorian.ly` auch eine Definition für \virgula und \caesura.

Vordefinierte Befehle

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

Siehe auch


Notationsreferenz: [Atemzeichen], Seite 127.

Installierte Dateien: `gregorian.ly`.

Referenz der Interna: Abschnitt “BreathingSign” in Referenz der Interna.

Artikulationszeichen des Gregorianischen Chorals

Zusätzlich zu den Standardartikulationszeichen, wie sie im Abschnitt [Artikulationszeichen und Verzierungen], Seite 113, beschrieben werden, werden auch Artikulationszeichen für die Notation des Editio Vaticana-Stils zur Verfügung gestellt.

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

\ictus

\circulus

\semicirculus
Kapitel 2: Spezielle Notation

Siehe auch

Notationreferenz: [Artikulationszeichen und Verzierungen], Seite 113.

Bekannte Probleme und Warnungen
Einige Artikulationszeichen sind vertikal zu dich an den entsprechenden Notenköpfen gesetzt.

Augmentationspunkte (*morae*)
Verlängerungspunkte, auch als *morae* bezeichnet, werden mit der Musikfunktion \texttt{\augmentum} hinzugefügt. Es handelt sich um eine eigenständige Funktion und nicht um einen Präfix, der zu einer Note gehört. Die Funktion wirkt sich nur auf den direkt vorhergehenden musikalischen Ausdruck aus. Das heißt, dass \texttt{\augmentum \texttt{\virga c}} keine sichtbare Wirkung hat. Anstelle dessen sollte geschrieben werden: \texttt{\texttt{\virga \augmentum c}} oder \texttt{\augmentum \{\virga c\}}. Man kann \texttt{\augmentum \{a g\}} als Kurznotation für \texttt{\augmentum a \augmentum g} schreiben.

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

Siehe auch

Notationsreferenz: [Atemzeichen], Seite 127.
Referenz der Interna: Abschnitt “BreathingSign” in Referenz der Interna.

Ligaturen der gregorianischen Quadratnotation

Die Unterstützung für gregorianische Neumen wird aktiviert, indem man mit \texttt{\include} die Datei \texttt{gregorian.ly} am Anfang der Quelldatei aktiviert. Damit werden zusätzliche Befehl zur Verfügung gestellt, mit denen man die Neumensymbole des Chorals produzieren kann.
Notenköpfe können verändert und/bzw. verbunden werden.

- Die Form des Notenkopfes kann verändert werden, indem man vor die Noten folgende Befehle schreibt: \virga, \stropha, \inclinatum, \auctum, \ascendens, \descendens, \oriscus, \quilisma, \deminutum, \cavum, \linea.

- Eigentliche Ligaturen (also Noten, die miteinander verbunden sind), werden erstellt, indem man einen der verbindenden Befehle, \pes oder \flexa für Aufwärts- bzw. Abwärtsbewegung, zwischen die zu verbindenden Noten setzt.

Eine Notenbezeichnung ohne jeglichen Modifikator produziert ein punctum. Alle anderen Neumen, auch einzelne Noten-Neumen mit einer anderen Form als der Virga werden generell als Ligaturen betrachtet und deshalb von den Zeichen \[ ... \] eingeklammert werden.

Einzelne Noten-Neumen:

- Das punctum ist die grundlegende Notenform (im Vaticana-Stil: ein Quadrat mit gebogenen Ober- und Unterkanten). Zusätzlich gibt es auch noch das oblique punctum inclinatum, das mit dem Präfix \inclinatum erstellt wird. Das normale punctum kann durch \cavum verändert werden, wodurch eine hohle Note erstellt wird, und durch \linea, wodurch vertikale Linien zu den Seiten der Note gezogen werden.

- Die virga hat einen absteigenden Hals auf der rechten Seite. Sie wird durch den Modifikator \virga erstellt.

Ligaturen

Anders als in anderen Neumennotationssystemen, wird das typographische Aussehen einer Ligatur nicht durch Eingabebefehle direkt vorgegeben, sondern richtet sich nach bestimmten Darstellungsregeln, die durch die musikalische Bedeutung bestimmt werden. Eine Ligatur mit drei Noten beispielsweise, mit der Form tief-hoch-tief, wie etwa \[ a \pes b \flexa g \], ergibt einen Torculus, der aus drei Punctum-Köpfen besteht, während die Form hoch-tief-hoch, wie etwa \[ a \flexa g \pes b \], einen Porrectus mit einer gebogenen Flexa und nur einem Punctum-Kopf ergibt. Es gibt keinen Befehl, mit dem explizit eine gebogene Flexa gesetzt werden können; die Entscheidung, wann eine derartige Form im Notenbild vorkommen soll, wird durch die musikalische Bedeutung der Noten vorgegeben. Die Idee hinter dieser Art der Eingabe ist es, dass der musikalische Inhalt von der graphischen Ausgabe getrennt wird. Dadurch wird es möglich, die gleiche Quelldatei zu benutzen, um beispielsweise die Noten in einem anderen Stil darzustellen.

Liquescente Neumen

Eine weitere Hauptkategorie der Notation von gregorianischem Choral sind die sogenannten liquezenten Neumen. Sie werden unter bestimmten Umständen am Ende einer Silbe eingesetzt, die auf einen „liquezenten“ Buchstaben endet (das sind die Konsonanten, die eine Tonhöhe haben können, also die Nasale, l, r, v, j und ihre diphtongalen Entsprechungen). Liquescente Neumen werden also nie alleine eingesetzt (auch wenn sie isoliert produziert werden können) und treten immer am Ende einer Silbe auf.

Liquescente Neumen werden graphisch auf zwei Arten dargestellt: mit einer kleineren Note oder indem die Hauptnote nach oben bzw. unten „gedreht“ wird. Die erste Darstellungsweise erreicht man, indem einen normalen pes oder flexa schreibt und dann die Form der zweiten Note verändert: \[ a \pes \deminutum b \]. Die zweite Darstellungsweise erreicht man, indem die Form einer einzelnen Neume mit \auctum und einem der Richtungsanzeiger \descendens bzw. \ascendens versieht: \[ \auctum \descendens a \].

Spezielle Zeichen

Eine dritte Kategorie besteht aus einer kleinen Anzahl an Zeichen mit einer besonderen Bedeutung: die quilisma, der oriscus und der strophicus. Sie werden notiert, indem man vor die entsprechende Note den Modifikator \quilisma, \oriscus oder \strophica schreibt.
Im Grunde kann innerhalb der Ligaturbegrenzer \[ und \] eine beliebige Anzahl aus Notenköpfen eingefügt werden und Prämisse wie \pes, \flexa, \virga, \inclinatum usw. können beliebig untereinander kombiniert werden. Der Einsatz der Regeln, mit denen die Ligaturen konstruiert werden, wird entsprechend angepasst. Auf diese Art kann eine unendliche Anzahl an Ligaturen erstellt werden.

Die Benutzung der Notationszeichen folgt allerdings bestimmten Regeln, die nicht von LilyPond überprüft werden. Die quilisma beispielsweise findet sich immer als mittlere Note einer aufsteigenden Ligatur und fällt üblicherweise auf einen Halbtonschnitt, aber es ist durchaus möglich, wenn auch nicht richtig, eine Quilisma bestehend aus einer Note zu notieren.

Neben den Notenformen definiert die Datei gregorian.ly auch die Befehle \versus, \responsum, \ij, \iiij, \Ij und \IIJ, mit denen die entsprechenden Zeichen, etwa für den Text oder als Abschnittsmarkierung erstellt werden können. Diese Befehl benutzen bestimmte Unicode-Zeichen und funktionieren nur, wenn eine Schriftart vorhanden ist, die diese Zeichen unterstützt.

In der folgenden Tabelle wird eine begrenzte, aber dennoch repräsentative Anzahl an Ligaturen der Neumennotation dargestellt, denen Fragmente beigefügt sind, die die Notation in LilyPond zeigen. Die Tabelle basiert auf der erweiterten Neumentabelle des zweiten Bands des Anti phonale Romanum (Liber Hymnarius), 1983 von den Mönchen von Solse mes herausgegeben. Die erste Spalte zeigt die Bezeichnungen der Ligaturen, fett für die Normalform, kursiv für die liquescente Form. Die dritte Spalte zeigt Code-Schnipsel, mit denen die Ligatur notiert werden kann, wobei die Noten g, a und b als Tonhöhen eingesetzt werden.

**Neumen aus einzelnen Noten**

<table>
<thead>
<tr>
<th>Grundform und liquescente Form</th>
<th>Ausgabe</th>
<th>LilyPond-Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Punctum</strong></td>
<td>[ b ]</td>
<td>[ b ]</td>
</tr>
<tr>
<td></td>
<td>[ cavum b ]</td>
<td>[ cavum b ]</td>
</tr>
<tr>
<td></td>
<td>[ linea b ]</td>
<td>[ linea b ]</td>
</tr>
<tr>
<td><strong>Punctum Auctum Ascendens</strong></td>
<td>[ auctum \ ascendens b ]</td>
<td>[ auctum \ ascendens b ]</td>
</tr>
<tr>
<td><strong>Punctum Auctum Descendens</strong></td>
<td>[ auctum \ descendens b ]</td>
<td>[ auctum \ descendens b ]</td>
</tr>
</tbody>
</table>
Punctum inclinatum \[
\text{\textbackslash inclinatum\ b}\]

Punctum Inclinatum Auctum \[
\text{\textbackslash inclinatum\ \textbackslash auctum\ b}\]

Punctum Inclinatum Parvum \[
\text{\textbackslash inclinatum\ \textbackslash diminutum\ b}\]

Virga

\[
\text{\textbackslash flexa\ g}\]

Ligaturen aus zwei Noten

Clivis vel Flexa \[
\text{\textbackslash flexa\ g}\]

Clivis Aucta Descendens \[
\text{\textbackslash flexa\ \textbackslash auctum\ \textbackslash descendens\ g}\]

Clivis Aucta Ascendens \[
\text{\textbackslash flexa\ \textbackslash auctum\ \textbackslash ascendens\ g}\]

Cephalicus \[
\text{\textbackslash flexa\ \textbackslash diminutum\ g}\]

Podatus/Pes \[
\text{\textbackslash pes\ b}\]
Kapitel 2: Spezielle Notation

*Pes Auctus Descendens*
\[ g \pes \auctum \descendens b \]

*Pes Auctus Ascendens*
\[ g \pes \auctum \ascendens b \]

*Epiphonus*
\[ g \pes \deminutum b \]

*Pes Initio Debilis*
\[ \deminutum g \pes b \]

*Pes Auctus Descendens Initio Debilis*
\[ \deminutum g \pes \auctum \descendens b \]

Ligaturen mit mehr als zwei Noten

*Torculus*
\[ a \pes b \flexa g \]

*Torculus Auctus Descendens*
\[ a \pes b \flexa \auctum \descendens g \]

*Torculus Deminutus*
\[ a \pes b \flexa \deminutum g \]

*Torculus Initio Debilis*
\[ \deminutum a \pes b \flexa g \]
Kapitel 2: Spezielle Notation

Torculus Auctus Descendens Initio Debilis
\[ \text{\deminutum a} \text{\pes b} \text{\flexa \auctum \descendens g} \]

Torculus Deminutus Initio Debilis
\[ \text{\deminutum a} \text{\pes b} \text{\flexa \deminutum g} \]

Porrectus
\[ \text{\a flexa g} \text{\pes b} \]

Porrectus Auctus Descendens
\[ \text{\a flexa g} \text{\pes \auctum \descendens b} \]

Porrectus Deminutus
\[ \text{\a flexa g} \text{\pes \deminutum b} \]

Climacus
\[ \text{\virga b} \text{\inclinatum a} \text{\inclinatum g} \]

Climacus Auctus
\[ \text{\virga b} \text{\inclinatum a} \text{\inclinatum \auctum g} \]

Climacus Deminutus
\[ \text{\virga b} \text{\inclinatum a} \text{\inclinatum \deminutum g} \]

Scandicus
\[ \text{g} \text{\pes a} \text{\virga b} \]
Kapitel 2: Spezielle Notation

*Scandicus Auctus Descendens*
\[
\text{\g\pes a \pes \auctum \descendens b} \]

*Scandicus Deminutus*
\[
\text{\g\pes a \pes \deminutum b} \]

**Special Signs**

*Quilisma*
\[
\text{\g \pes \quilisma a \pes b} \]

*Quilisma Pes Auctus Descendens*
\[
\text{\quilisma \g \pes \auctum \descendens b} \]

*Oriscus*
\[
\text{\oriscus b} \]

*Pes Quassus*
\[
\text{\oriscus \g \pes \virga b} \]

*Pes Quassus Auctus Descendens*
\[
\text{\oriscus \g \pes \auctum \descendens b} \]

*Salicus*
\[
\text{\g \oriscus a \pes \virga b} \]

*Salicus Auctus Descendens*
\[
\text{\g \oriscus a \pes \auctum \descendens b} \]
Kapitel 2: Spezielle Notation

(Apo)stropha
\[ \stropha b \]

Stropha Aucta
\[ \stropha \auctum b \]

Bistropha
\[ \stropha b \stropha b \]

Tristropha
\[ \stropha b \stropha b \stropha b \]

Trigonus
\[ \stropha b \stropha b \stropha a \]

Vordefinierte Befehle
Folgende Notenpräfixe sind unterstellt: \virga, \stropha, \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.

Siehe auch
Glossar: Abschnitt “ligature” in Glossar.

Notationreferenz: ⟨undefined⟩ [Ligaturen der gregorianischen Quadratnotation], Seite ⟨undefined⟩, [Weiße Mensuralligaturen], Seite 420, [Ligaturen], Seite 412.

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.

2.9.5 Kiever Quadratnotation setzen
Kiever Kontexte

Wie auch für die Mensural- und Gregorianische Notation können die Kontexte KievanVoice und KievanStaff eingesetzt werden, um Noten der Kiever Quadratnotation zu setzen. Diese Kontexte initialisieren die benötigten Kontexteigenschaften und Grob-Eigenschaften mit den richtigen Werten, sodass man sofort den Choral notieren kann:

```plaintext
\% Font settings for Cyrillic
\paper {
  #(define fonts
    (set-global-fonts
      #:roman "Linux Libertine O,serif"
    ))
}

\score {
  \new KievanVoice = "melody" \transpose c c' {
    \cadenzaOn
    c4 c c c c2 b,\longa
    \bar "kievan"
  }
  \new Lyrics \lyricsto "melody" {
    -- -- -- -- .
  }
  \}
}
```

Господи помилуй.

Siehe auch

Glossar: Abschnitt “kievan notation” in Glossar.

Bekannte Probleme und Warnungen

LilyPond unterstützt Kiever Notation des Synodischen Stils, welcher im Korpus der Gesangsbücher eingesetzt wurde, die durch Russische Heilige Synode 1910 gedruckt wurden und neuerdings durch das Moskauer Patriarchat Verlagshaus neu herausgegeben wurden. LilyPond kann nicht die älteren (seltener) Formen der Kiever Notation setzen, mit denen in Galizien rusinischer Choral notiert worden ist.

Kiever Schlüssel

Es gibt nur einen Schlüssel in der Kiever Notation (der Tse-fa-ut-Schlüssel). Er bezeichnet die Position von c:

```plaintext
\clef "kievan-do"
\kievanOn
\kiclenOn
\c'
```

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure.png}
\caption{Example of Kiever Schlüssel notation.}
\end{figure}
Kapitel 2: Spezielle Notation 434

Siehe auch

Glossar: Abschnitt “kievan notation” in Glossar, Abschnitt “clef” in Glossar.
Notationsreferenz: [Notenschlüssel], Seite 16.

Kiever Notenköpfe

Für die Kiever Notation muss der richtige Notenkopfstil gewählt werden. Die erreicht man, indem man die style-Eigenschaft des NoteHead-Objekts auf kievan setzt.

Die Kiever Schlussnote, welche am Ende eines Stückes gesetzt wird, kann gewählt werden, indem man die Notendauer \longa einsetzt. Das Kiever Rezitativzeichen, welches die Rezitation auf einer Tonhöhe anzeigt, kann gesetzt werden, indem die Notendauer \breve notiert wird. Folgendes Beispiel demonstriert die unterschiedlichen Notenköpfe:

\autoBeamOff
\cadenzaOn
\kievanOn
b'1 b'2 b'4 b'8 b'\breve b'\longa
\kievanOff
b'2

Siehe auch

Glossar: Abschnitt “kievan notation” in Glossar, Abschnitt “note head” in Glossar.
Notationsreferenz: Abschnitt A.9 [Notenkopfstile], Seite 646.

Bekannte Probleme und Warnungen

LilyPond bestimmt automatisch die Richtung eines Halses. Für Gesang in der Quadratnotation zeigen die Hälse jedoch alle in die gleiche Richtung innerhalb eines Melismas. Das kann man manuell erreichen, indem man die direction-Eigenschaft des Stem-Objekts setzt.

Kiever Versetzungszeichen

Der Versetzungszeichenstil kievan wird durch die Eigenschaft glyph-name-alist des Accidental-Grobs ausgewählt. Dieser Stil stellt ein Kreuz und ein B-Zeichen zur Verfügung, die sich von den Standardzeichen unterscheiden. Es gibt kein Auflösungszeichen in der Kiever Notation. Das Kreuz wird in der Synodalen Musik nicht eingesetzt, kann aber in früheren Manuskripten auftreten. Es wurde vor allem der Vollständigkeit halber eingesetzt.

\clef "kievan-do"
\override Accidental.glyph-name-alist =
#alteration-kievan-glyph-name-alist
bes' dis'

Siehe auch

Glossar: Abschnitt “kievan notation” in Glossar, Abschnitt “accidental” in Glossar.
Notationsreferenz: [Versetzungszeichen], Seite 5, [Automatische Versetzungszeichen], Seite 24, Abschnitt A.8 [Die Emmentaler-Schriftart], Seite 625.
### Kiever Taktstriche

Eine dekorative Figur wird üblicherweise am Ende eines Musikstücks der Kiever Notation gesetzt, was man als Kiever Schlussstrich bezeichnen kann. Es wird gesetzt mit \( \texttt{\bar "k"} \).

\begin{verbatim}
\clef "kievan-do"
\kievanOn
\bar "k"
\end{verbatim}

Siehe auch

Abschnitt 1.2.5 [Takte], Seite 92, Abschnitt A.8 [Die Emmentaler-Schriftart], Seite 625.

#### 2.9.6 Musiksatz Alter Musik in der Praxis – Szenarien und Lösungen

Wenn man mit Alter Notation zu tun hat, fallen oft Aufgaben an, die in der modernen Notation nicht vorkommen, für welche LilyPond geschaffen wurde. In diesem Abschnitt sollen darum einige praktische Problemstellungen und Lösungsvorschläge dargestellt werden. Dabei handelt es sich um:

- wie man Incipite in modernen Editionen von Mensuralnotation notieren kann (d.h. ein kleiner Abschnitt vor der eigentlichen Partitur, der die Originalnotenformen darstellt),
- wie man \textit{Mensurstriche} einstellt, mit denen oft moderne Transkriptionen polyphoner Musik notiert werden,
- wie man den gregorianischen Choral mit moderner Notation darstellt und
- wie man sowohl ein Mensuralnotationsbild als auch eine moderne Edition aus der selben Quelle erstellt.

### Incipite

In Arbeit.

### Mensurstriche

Als \textit{Mensurstriche} wird ein Notenlayout bezeichnet, in dem die Taktlinien nicht auf den Systemen, sondern nur zwischen Systemen gezogen werden. Damit soll signalisiert werden, dass das Original keine Takteinteilung besessen hat und etwa Synkopen nicht über Taktlinien hinweg aufgeteilt werden müssen, während man sich dennoch an den Taktlinien rhythmisch orientieren kann.

Das Mensurstriche-Layout, in welchem die Taktlinien nicht auf den Systemen, sondern zwischen den Systemen gezogen werden, kann mit einer \texttt{StaffGroup} anstelle von \texttt{ChoirStaff} erreicht werden. Die Taktlinien auf den Systemen werden mit der \texttt{transparent}-Eigenschaft ausgelöscht.

\begin{verbatim}
\global = { \
  \hide Staff.BarLine\n  s1 s\n  \% the final bar line is not interrupted\n  \undo \hide Staff.BarLine\n  \bar "|."\n}\n\new StaffGroup \relative c'' { \n  \<< \new Staff { \<< \global { c1 c } >> } \n}\n\end{verbatim}
Gregorianischen Choral transkribieren

Gregorianischer Choral kann mit einigen einfachen Einstellungen in moderner Notation notiert werden.

Häse. Häse können meistens weggelassen werden, was geschieht, indem man den Stem_engraver aus dem Stimmenkontext entfernt:

```
\layout {
  ...
  \context {
    \Voice
    \remove "Stem_engraver"

  }
}
```

In einigen Transkriptionsstilen werden jedoch teilweise Häse eingesetzt, um etwa den Übergang von einem Einton-Rezitativ zu einer melodischen Geste anzuzeigen. In diesem Fall können Häse entweder mit \hide Stem unsichtbar gemacht werden oder mit \override Stem.length = #0 auf die Länge von 0 reduziert werden. Die Häse müssen dann wieder an den entsprechenden Stellen mit \once \override Stem.transparent = ##f sichtbar gemacht werden (siehe auch Beispiel unten). Wenn Häse eingesetzt werden, die Fähnchen haben, muss zusätzlich auch noch \hide Flag eingestellt werden.

Takt. Für Gesang ohne Metrum gibt es einige Alternativen.

Der Time_signature_engraver kann aus dem Staff-Kontext entfernt werden, ohne dass es negative Seitenefekte gäbe. Alternativ kann er durchsichtig gemacht werden, dabei entsteht aber ein leerer Platz zu Beginn der Noten an der Stelle, wo normalerweise die Taktangabe stehen würde.

In vielen Fällen ergibt \set Score.timing = ##f gute Ergebnisse. Eine andere Möglichkeit ist es, \cadenzaOn und \cadenzaOff zu benutzen.

Um Taktstriche zu entfernen, kann man radikal den Bar_engraver aus dem Staff-Kontext entfernen. Wenn man ab und zu einen Taktstrich braucht, sollten die Striche nur mit \hide BarLine unsichtbar gemacht werden.

Oft werden Rezitativtöne mit einer Brevis angezeigt. Der Text für die Rezitativnote kann auf zwei Arten notiert werden: entweder als einzelne, links ausgerichtete Silbe:

```
\include "gregorian.ly"
chant = \relative {
  \clef "G_8"
  c\breve c4 b4 a c2 c4 \divisioMaior
  c\breve c4 c f, f \finalis
}
```
verba = \lyricmode {
  \once \override LyricText.self-alignment-X = #-1
  "Noctem quietam et" fi -- nem per -- fec -- tum
  \once \override LyricText.self-alignment-X = #1
  "concedat nobis Dominus" om -- ni -- po -- tens.
}
\score {
  \new Staff <<
  \new Voice = "melody" \chant
  \new Lyrics = "one" \lyricsto melody verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \hide Stem
    }
  }
}

\include "gregorian.ly"
chant = \relative {
  \clef "G_8"
  \set Score.timing = ##f
c'\breve \hide NoteHead c c c c c
  \undo \hide NoteHead
  \override Stem.transparent = ##f \stemUp c4 b4 a
  \hide Stem c2 c4 \divisioMaior
  c'\breve \hide NoteHead c c c c c c
  \undo \hide NoteHead c4 c f, f \finalis
}
verba = \lyricmode {
  No -- ctem qui -- e -- tam et fi -- nem per -- fec -- tum
  con -- ce -- dat no -- bis Do -- mi -- nus om -- ni -- po -- tens.
}
\score {
  \new Staff <<
Eine andere übliche Situation ist die Transkription von neumatiscem oder melismatischem Gesang, d.h. Gesang, der eine unterschiedliche Anzahl von Noten pro Silbe hat. In diesem Fall sollen die Silbengruppen üblicherweise deutlich voneinander getrennt gesetzt werden, oft auch die Untergruppen eines längeren Melismas. Eine Möglichkeit, das zu erreichen, ist es, eine feste Taktart, etwa 1/4, zu benutzen und dann jeder Silbe oder Notengruppe einen ganzen Takt zuzuweisen, u.U. mit Hilfe von Triolen und kleinen Notenwerten. Wenn die Taktstriche und alle anderen rhythmischen Anweisungen unsichtbar gemacht werden, und der Platz um die Taktstriche vergrößert wird, ergibt sich eine recht gute Repräsentation der Originalnotation.

Damit Silben mit unterschiedlicher Länge (etwa "-ri" und "-rum") die Silbengruppen nicht ungleichmäßig aufweiten, kann die 'X-extent-Eigenschaft des LyricText-Objekts auf einen festen Wert gesetzt werden. Eine andere Möglichkeit wäre es, die Silben als Textbeschriftung einzufügen. Wenn weitere horizontale Anpassungen nötig sind, können sie mit unsichtbaren (s)-Noten vorgenommen werden.

spiritus = \relative { 
  \time 1/4 
  \override Lyrics.LyricText.X-extent = #'(0 . 3) 
  d'4 \tuplet 3/2 { f8 a g } g a a4 g f8 e  
  d4 f8 g g8 d f g a f4 g8 a a4 s 
  \tuplet 3/2 { g8 f d } e f g a g4 
} 

spirLyr = \lyricmode {
  Spi -- ri -- _ _ tus _ Do -- mi -- ni _ re -- ple -- _ vit _
  or -- _ bem _ ter -- ra -- _ rum, al -- _ le -- _ lu
  -- _ ia.
}

\score { 
  \new Staff <<
    \new Voice = "chant" \spiritus 
    \new Lyrics = "one" \lyricsto "chant" \spirLyr
  >>
  \layout {
    \context {
      \Staff

Noctem qui etam et finem perfectum concedat nobis Dominus omnipotens.
Alte und moderne Edition aus einer Quelldatei
In Arbeit.

Herausgeberische Anmerkungen
In Arbeit.

2.10 Weltmusik
Dieser Abschnitt soll Besonderheiten der Notation aufzeigen, die insbesondere relevant sind, um Musik nicht-westlicher Tradition zu notieren.

2.10.1 Übliche Notation für nichteuropäische Musik
Dieser Abschnitt zeigt, wie man Partituren erstellt, die nicht der europäischen klassischen Musiktradition angehören.

Erweiterung von Notation und Stimmungssystemen
Die klassische Standardnotation wird üblicherweise zur Notation verschiedener Musikarten benutzt und ist nicht auf die „klassische Musik“ beschränkt. Diese Notation wird behandelt in Abschnitt 1.1.1 [Tonhöhen setzen], Seite 1, und die unterschiedlichen Notenbezeichnungen, die eingesetzt werden können, finden sich in [Notenbezeichnungen in anderen Sprachen], Seite 7.

Viele nicht-europäische Musik (und auch manche europäische Volksmusik) benutzt jedoch alternative oder erweiterte Skalen (Tonleitern), die man nicht mit der normalen westlichen Notation notieren kann.

In einigen Fällen wird die klassische Notation dennoch benutzt, wobei man die Tonhöhenunterschiede implizit mitliest. Beispielsweise arabische Musik wird mit normalen Halb- und Vierteltonversetzungsschlägen notiert und die exakte Tonhöhe (die etwas von der notierten abweichen kann) dann aus dem Kontext erschlossen. Italienische Notenbezeichnungen werden normalerweise benutzt, und die Init-Datei arabic.ly stellt eine Anzahl an Makros zur Verfügung, die die Standardnotation erweitern. Siehe auch Abschnitt 2.10.2 [Arabische Musik], Seite 440.

Um Dateien wie arabic.ly oder makam.ly zu finden, siehe Abschnitt “Mehr Information” in Handbuch zum Lernen.

Ausgewählte Schnipsel

_Makam-Beispiel_

Makam ist eine türkische Melodie, in der 1/9-Tonabstände eingesetzt werden. Sehen Sie sich die Initialisierungsdatei makam.ly für weiter Information zu Tonhöhenbezeichnungen und Alterationen an (siehe Handbuch zum Lernen 2.19.84, 4.6.3 Weitere Information zu Hinweisen, wo diese Datei gespeichert ist).

```
\relative c' {
    \set Staff.keyAlterations = #'((6 . ,(- KOMA)) (3 . ,BAKIYE))
    c4 cc db fk
    gbm4 gfc gfb efk
    fk4 db cc c
}
```

Siehe auch


Notationsreferenz: Abschnitt 1.1.1 [Tonhöhen setzen], Seite 1, [Notenbezeichnungen in anderen Sprachen], Seite 7, Abschnitt 2.10.2 [Arabische Musik], Seite 440, Abschnitt 2.10.3 [Türkische klassische Musik], Seite 445.

2.10.2 Arabische Musik

Dieser Abschnitt zeigt Möglichkeiten, wie arabisch Musik notiert werden kann.

Referenz für arabische Musik

Arabische Musik wurde bisher vor allem mündlich tradiert. Wenn Musik transkribiert wird, handelt es sich meistens um ein Gerüst, auf dem der Musiker eigene Improvisationen ausführt. Mehr und mehr wird die westliche Notation mit einigen Veränderungen benutzt, um die arabische Musiktradition weiterzugeben und zu konservieren.

Einige Elemente der westlichen Notation wie etwa die Transkription von Akkorden oder eigenständige Stimmen werden für die traditionellen arabischen Noten nicht benötigt. Es gibt allerdings einige andere Probleme, wie etwa die Notwendigkeit, Zwischenintervalle zu notieren, die sich irgendwo zwischen einem Halbton und einem Ganzton befinden. Daneben werden auch
Kapitel 2: Spezielle Notation

die westlichen Halb- und Ganztöne eingesetzt. Es muss auch möglich sein, eine große Anzahl an maqam (Modi) der arabischen Musik zu bezeichnen und zu gruppieren.

Üblicherweise müssen Mikrotöne in der arabischen Musik nicht präzise notiert werden.
Einige Bereiche, die für die arabische Notation wichtig sind, sind an anderer Stelle behandelt:
• Notenbezeichnungen und Versetzungszeichen (inklusive Vierteltöne) können angepasst werden, wie behandelt in Abschnitt 2.10.1 [Übliche Notation für nichteuropäische Musik], Seite 439.
• Zusätzliche Taktarten können erstellt werden, siehe [Tonartbezeichnung], Seite 20.
• Komplexe Taktarten erfordern evtl., dass Noten manuell gruppiert werden, wie gezeigt in [Manuelle Balken], Seite 88.
• Takasim, rhythmisch freie Improvisationen, können ohne Taktlinien notiert werden, siehe hierzu [Musik ohne Metrum], Seite 69.

Siehe auch

Notationsreferenz: Abschnitt 2.10.1 [Übliche Notation für nichteuropäische Musik], Seite 439, [Tonartbezeichnung], Seite 20, [Manuelle Balken], Seite 88.

Arabische Notenbezeichnungen

An der arabischen Tradition orientierte Notenbezeichnungen können sehr land sein und eignen sich daher nicht gut für die Notation von Musik. Sie werden nicht benutzt. Englische Notenbezeichnungen hingegen sind in der arabischen Musikerziehung recht unbekannt, weshalb italienische Notenbezeichnungen (do, re, mi, fa, sol, la, si) eingesetzt werden. Modifikatoren (Versetzungszeichen) können auch benutzt werden. Italienische Notenbezeichnungen finden sich erläutert in [Notenbezeichnungen in anderen Sprachen], Seite 7, die Benutzung der normalen europäischen Notation für nichteuropäische Musik findet sich erläutert in Abschnitt 2.10.1 [Übliche Notation für nichteuropäische Musik], Seite 439.
Hier ein Beispiel der arabischen rast-Tonleiter:

\include "arabic.ly"
\relative {
  do' re misb fa sol la sisb do sisb la sol fa misb re do
}

\set Staff.extraNatural = ##f
\relative {
  dod' dob dosd \dwn dob dobsb dosdsd do do
}

Das Symbol für das Halb-B sieht anders aus als das Symbol, was üblicherweise in arabischer Notation benutzt wird. Das \dwn-Symbol, das in der Datei arabic.ly definiert ist, kann als ein Workaround eingesetzt werden, wenn es notwendig ist, das arabische Symbol zu benutzen. Das Aussehen des Halb-Bs in den Vorzeichen kann mit dieser Methode nicht verändert werden.
Arabische Tonarten


Ein Maqam kann die Tonart der Gruppe benutzen, zu der er gehört, oder die einer benachbarten Gruppe. Zusätzlich können verschiedene Versetzungszeichen in den Noten markiert werden.

Um also etwa die Tonart des Maqams „muhayer“ folgendermaßen notiert:

\key re \bayati

re ist die Tonhöhe für den „muhayer“-Maqam und bayati ist die Bezeichnung des Basismaqams der Gruppe.

Während die Vorzeichen eine Gruppe anzeigen, wird meistens der eigentliche Maqam im Titel definiert. In diesem Beispiel müsste also der „muhayer“-Maqam im Titel erscheinen.

Andere Maqams derselben Bayati-Gruppe, wie in der Tabelle unten gezeigt ((bayati, hussaini, saba und ushaq) können auf die gleiche Weise notiert werden. Sie sind alle Variationen des Grundmaqams Bayati. Sie unterscheiden sich üblicherweise vom grundlegenden Maqam in ihrem oberen Tetrachord oder in bestimmten Einzelheiten, die aber nicht ihre eigentliche Qualität verändern.

Der andere Maqam der gleichen Gruppe (Nawa) ist mit bayati durch eine Modulation verkümpft, deren Grundton in der Tabelle angezeigt wird, wenn es sich um einen Maqam handelt, der eine Modulation eines anderen Maqams darstellt. Nawa kann folgendermaßen notiert werden:

\key sol \bayati

In der arabischen Musik ist ein Begriff wie bayati, der eine Maqam-Gruppe bezeichnet, gleichzeitig auch selber ein Maqam, meistens der häufigste dieser Gruppe.

Hier ist eine Möglichkeit, Maqams zu gruppieren, womit die häufigsten Maqams bestimmten Vorzeichen zugeordnet werden:

<table>
<thead>
<tr>
<th>Maqam-Gruppe</th>
<th>Vorzeichen</th>
<th>Finalis</th>
</tr>
</thead>
<tbody>
<tr>
<td>ajam</td>
<td>major</td>
<td>sib</td>
</tr>
<tr>
<td>bayati</td>
<td>bayati</td>
<td>re</td>
</tr>
<tr>
<td>hijaz</td>
<td>kurd</td>
<td>re</td>
</tr>
<tr>
<td>iraq</td>
<td>iraq</td>
<td>sib</td>
</tr>
<tr>
<td>kurd</td>
<td>kurd</td>
<td>re</td>
</tr>
<tr>
<td>mahawand</td>
<td>minor</td>
<td>do</td>
</tr>
<tr>
<td>nakriz</td>
<td>minor</td>
<td>do</td>
</tr>
<tr>
<td>rast</td>
<td>rast</td>
<td>do</td>
</tr>
<tr>
<td>sikah</td>
<td>sikah</td>
<td>misb</td>
</tr>
</tbody>
</table>

Ausgewählte Schnipsel

Untypische Tonarten

Der üblicherweise benutzte key-Befehl setzt die keySignature-Eigenschaft im Staff-Kontext.
Um untypische Tonartenvorzeichen zu erstellen, muss man diese Eigenschaft direkt setzen. Das Format für den Befehl ist eine Liste:
\set Staff.keySignature = #`(((Oktave . Schritt) . Alteration) ((Oktave . Schritt) . Alteration) ...) wobei für jedes Element in der Liste Oktave die Oktave angibt (0 ist die Oktave vom eingestrichenen C bis zum eingestrichenen H), Schritt gibt die Note innerhalb der Oktave an (0 heißt C und 6 heißt H), und Alteration ist ,SHARP ,FLAT ,DOUBLE-SHARP usw. (Beachte das beginnende Komma.)

Alternativ kann auch jedes Element der Liste mit dem allgemeineren Format (Schritt . Alteration) gesetzt werden, wobei dann die Einstellungen für alle Oktaven gelten.

Hier ein Beispiel einer möglichen Tonart für eine Ganztonleiter:

```latex
\include "arabic.ly"
\relative do' {
\set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
(1 . ,SEMI-FLAT)
(2 . ,FLAT)
(5 . ,FLAT)
(6 . ,SEMI-FLAT))
\set Staff.extraNatural = ##f
re reb \dwn reb resd
dod dob dosd \dwn dob |
dobsb dodsd do do |
```
Bei Improvisation oder taqasim, die zeitlich frei gespielt werden, kann die Taktart ausgelassen werden und \cadenzaOn kann eingesetzt werden. Es kann nötig sein, den Versetzungszeichenstil anzupassen, weil sonst die Versetzungszeichen nur einmal ausgegeben werden, da keine Taktlinien gesetzt sind. Hier ein Beispiel, wie der Beginn einer hijaz-Improvisation aussehen könnte:

\include "arabic.ly"

\relative sol' {
  \key re \kurd
  \accidentalStyle forget
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}

Siehe auch

Glossar: Abschnitt “semai” in Glossar, Abschnitt “taqasim” in Glossar.
Notationsreferenz: [Manuelle Balken], Seite 88, [Automatische Balken], Seite 77, [Musik ohne Metrum], Seite 69, [Automatische Versetzungszeichen], Seite 24, [Einstellung von automatischen Balken], Seite 80, [Taktangabe], Seite 60.

Arabische Notenbeispiele
Hier eine Vorlage, welche den Beginn eines türkischen Semai benutzt, der in der arabischen Musikerziehung oft herangezogen wird, um Besonderheiten der arabischen Musiknotation, wie etwa Zwischenintervalle und ungewöhnliche Modi, zu illustrieren.

\include "arabic.ly"
\score {
  \relative {
    \set Staff.extraNatural = ##f
    \set Staff.autoBeaming = ##f
    \key re \bayati
    \time 10/8
    re'4 re'8 re16 [misb re do] sissb [la sissb do] re4 r8
    do4 sissb8 misb16 [re do sissb] la [do sissb la] la4 r8
  }
  \header {
    title = "Semai Muhayer"
    composer = "Jamil Bek"
  }
}

\relative sol' {
  \key re \kurd
  \accidentalStyle forget
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}
Siehe auch


Weitere Literatur zur arabischen Musik


Es gibt auch einige Internetseiten, die Maqams erklären und teilweise auch Klangdateien zur Verfügung stellen:
- http://www.maqamworld.com/
- http://www.turath.org/

Die Maqam-Gruppierungen unterscheiden sich in einigen Details, auch wenn die allgemeinen Kriterien weithin anerkannt sind: gemeinsame untere Tetrachorde sowie Modulation.

2. Es gibt keine Übereinstimmung darüber, wie die Vorzeichen für bestimmte Maqams angegeben werden sollen. Oft wird eine Vorzeichenart für eine ganze Maqam-Gruppe verwendet, anstatt dass jeder Maqam eigene Vorzeichen hätte.

Lehrbücher für Oud, die arabische Laute, folgender Autoren enthalten Beispiele vor allem türkischer und arabischer Kompositionen:
- Charbel Rouhana
- George Farah
- Ibrahim Ali Darwish Al-masri

2.10.3 Türkische klassische Musik

Dieser Abschnitt zeigt Probleme, die bei der Notation von klassischer türkischer Musik auftreten können.

Verweise für türkische klassische Musik


Einige Probleme der Notation türkischer klassischer Musik sind wounders behandelt:
- Notenbezeichnungen und Versetzungszeichen finden sich in Abschnitt 2.10.1 [Übliche Notation für nichteuropäische Musik], Seite 439.

Türkische Notenbezeichnungen


Es bietet sich an, die normalen westlichen Noten auf dem Notensystem zu benutzen (also c, d, e ... ) angereichert mit besonderen Versetzungszeichen, die die Noten um 1/9, 4/9, 5/9 und 8/9 eines Ganztons erhöhen oder erniedrigen. Diese Versetzungszeichen sind definiert in der Datei makam.ly.
Die folgende Tabelle zeigt
• die Bezeichnung dieser besonderen Versetzungszeichen
• die Endung, die an die Note gehängt werden muss und
• die Tonhöhenveränderung als Bruch eines Ganztones.

<table>
<thead>
<tr>
<th>Versetzungszeichen</th>
<th>Endung</th>
<th>Tonhöhenveränderung</th>
</tr>
</thead>
<tbody>
<tr>
<td>b&quot;uy&quot;uk m&quot;ucenneb (Kreuz)</td>
<td>-bm</td>
<td>+8/9</td>
</tr>
<tr>
<td>k&quot;uc&quot;uk m&quot;ucenneb (Kreuz)</td>
<td>-k</td>
<td>+5/9</td>
</tr>
<tr>
<td>bakiye (Kreuz)</td>
<td>-b</td>
<td>+4/9</td>
</tr>
<tr>
<td>koma (Kreuz)</td>
<td>-c</td>
<td>+1/9</td>
</tr>
<tr>
<td>koma (B)</td>
<td>-fc</td>
<td>-1/9</td>
</tr>
<tr>
<td>bakiye (B)</td>
<td>-fb</td>
<td>-4/9</td>
</tr>
<tr>
<td>k&quot;uc&quot;uk m&quot;ucenneb (B)</td>
<td>-fk</td>
<td>-5/9</td>
</tr>
<tr>
<td>b&quot;uy&quot;uk m&quot;ucenneb (B)</td>
<td>-fbm</td>
<td>-8/9</td>
</tr>
</tbody>
</table>

Eine weitergehende Erklärung der Notation nichteuropäischer Musik findet sich in Abschnitt 2.10.1 [Übliche Notation für nichteuropäische Musik], Seite 439.

Siehe auch

Glossar: Abschnitt “makam” in Glossar, Abschnitt “makamlar” in Glossar.

Notationsreferenz: Abschnitt 2.10.1 [Übliche Notation für nichteuropäische Musik], Seite 439.
3 Allgemeine Eingabe und Ausgabe

Dieses Kapitel erläutert allgemeine Fragen zur Eingabe und Ausgabe von Notation mit LilyPond und weniger direkte Fragen der Notation.

3.1 Eingabestruktur

Das hauptsächliche Eingabeformat von LilyPond sind Textdateien. Üblicherweise werden diese Dateien mit der Endung .ly versehen.

3.1.1 Struktur einer Partitur

Eine \score-Umgebung muss einen einzelnen musikalischen Ausdruck beinhalten, der durch geschweifte Klammern begrenzt wird:

\score { 
... 
}

\score { c'4 c' c' c' }
{ c'4 c' c' c' }
{ d'4 d' d' d' }

Dieser einzelne musikalische Ausdruck kann beliebige Größe annehmen und andere musikalische Ausdrücke von beliebiger Komplexität beinhalten. Alle diese Beispiele sind musikalische Ausdrücke:

<< \new Staff { c'4 c' c' c' } \new Staff { d'4 d' d' d' } >>

{ \new GrandStaff << \new StaffGroup << \new Staff { \Flöte } \new Staff { \Oboe } >>}
Kapitel 3: Allgemeine Eingabe und Ausgabe

\new StaffGroup <<
  \new Staff { \GeigeI }
  \new Staff { \GeigeII }
>>

Kommentare bilden eine Ausnahme dieser Regel. (Andere Ausnahmen siehe Abschnitt 3.1.5 [Die Dateistruktur], Seite 451.) Sowohl einzeilige als auch Blockkommentare (eingegrenzt durch \%{ .. \%}) können an beliebiger Stelle einer Eingabedatei geschrieben werden. Sie können innerhalb oder außerhalb der \score-Umgebung vorkommen, und innerhalb oder außerhalb des einzelnen musikalischen Ausdrucks innerhalb der \score-Umgebung.

Denken Sie daran, dass auch eine Datei, die nur eine \score-Umgebung enthält, implizit in eine \book-Umgebung eingeschlossen wird. Eine \book-Umgebung in einer Eingabedatei produziert wenigstens eine Ausgabedatei, und standardmäßig wird der Name der Ausgabedatei aus dem Namen der Eingabedatei abgeleitet. fandangoforelephants.ly produziert also fandangoforelephants.pdf.

Zu weiteren Einzelheiten zu \book-Umgebungen siehe Abschnitt 3.1.2 [Mehrere Partituren in einem Buch], Seite 448, Abschnitt 3.1.3 [Mehrere Ausgabedateien aus einer Eingabedatei], Seite 449, und Abschnitt 3.1.5 [Die Dateistruktur], Seite 451.

Siehe auch


Notationsreferenz: Abschnitt 3.1.2 [Mehrere Partituren in einem Buch], Seite 448, Abschnitt 3.1.3 [Mehrere Ausgabedateien aus einer Eingabedatei], Seite 449, und Abschnitt 3.1.5 [Die Dateistruktur], Seite 451.

3.1.2 Mehrere Partituren in einem Buch

Eine Partitur kann mehrere musikalische Stücker und verschiedene Texte enthalten. Beispiele hierzu sind etwa eine Etüdenübersicht oder ein Orchesterstück mit mehreren Sätzen. Jeder Satz wird in einer eigenen \score-Umgebung notiert:
\score {
  ..Noten..
}

und Texte werden mit einer \markup-Umgebung geschrieben:
\markup {
  ..Text..
}

Alle Sätze und Texte, die in derselben .ly-Datei vorkommen, werden normalerweise in eine einzige Ausgabedatei gesetzt.
\score {
  ...
}
\markup {
  ...
}
\score {
  ...
}
Eine wichtige Ausnahme stellen Dokumente dar, die mit lilypond-book erstellt werden, für die Sie explizit \book-Umgebungen notieren müssen, weil sonst nur die erste \score-bzw. \markup-Umgebung angezeigt wird.

Der Kopfbereich für jedes Musikstück kann innerhalb der \score-Umgebung definiert werden. Die piece-(Stück)-Bezeichnung aus dieser \header-Umgebung wird vor jedem Satz ausgegeben. Die Überschrift für ein ganzes Buch kann innerhalb von \book notiert werden, aber wenn diese Umgebung fehlt, wird die \header-Umgebung genommen, die auf erster Ebene der Datei notiert ist.

\header {
    title = "Acht Miniaturen"
    composer = "Igor Stravinsky"
}
\score {
    ...
    \header { piece = "Romanze" }
}
\markup {
    ..Text der zweiten Strophe..
}
\markup {
    ..Text der dritten Strophe..
}
\score {
    ...
    \header { piece = "Menuetto" }
}

Stücke können innerhalb eines Buches mit \bookpart gruppiert werden. Derartige Buchabschnitte werden durch einen Seitenumbruch voneinander getrennt und können wie auch das ganze Buch selber mit einem Titel innerhalb einer \header-Umgebung beginnen.

\bookpart {
    \header {
        title = "Buchtitel"
        subtitle = "Erster Teil"
    }
    \score { ... }
    ...
}
\bookpart {
    \header {
        subtitle = "Zweiter Teil"
    }
    \score { ... }
    ...
}

3.1.3 Mehrere Ausgabedateien aus einer Eingabedatei

Wenn man mehrere Dateien aus einer einzigen Eingabedatei erstellt, stellt LilyPond sicher, dass keine der Ausgabedateien der vorhandenen \book-Umgebungen eine andere Ausgabedatei, etwa von der vorherigen \book-Umgebung, überschreibt.

Dies geschieht, indem ein Suffix an den Ausgabenamen für jede \book-Umgebung gehängt wird, die den Dateinamen der Eingabedatei als Grundlage nimmt. Das Standardverhalten ist es, einen Zahlen-Suffix für die Namen hinzuzufügen, die in Konflikt stehen. Der Code
\begin{verbatim}
\book{
  \score{...}
  \layout{...}
}
\book{
  \score{...}
  \layout{...}
}
\book{
  \score{...}
  \layout{...}
}
\end{verbatim}
produziert also
- eightminiatures.pdf,
- eightminiatures-1.pdf and
- eightminiatures-2.pdf.

### 3.1.4 Dateinamen der Ausgabedateien

LilyPond stellt die Möglichkeit zur Verfügung zu kontrollieren, welche Dateinamen für welche Back-ends benutzt werden sollen, wenn die Ausgabedateien erstellt werden.

\begin{verbatim}
\book{
  \bookOutputSuffix"Romanze"
  \score { ... }
  \layout { ... }
}
\book{
  \bookOutputSuffix"Menuetto"
  \score { ... }
  \layout { ... }
}
\book{
  \bookOutputSuffix"Nocturne"
  \score { ... }
  \layout { ... }
}
\end{verbatim}
Man kann auch einen anderen Dateinamen für die Ausgabedatei einer \book-Umgebung erstellen, indem man \bookOutputName-Angabe macht:
Kapitel 3: Allgemeine Eingabe und Ausgabe

Die obige Datei produziert folgende Ausgabedateien:
- Romanze.pdf
- Menuetto.pdf
- Nocturne.pdf

3.1.5 Die Dateiestruktur

Eine .ly-Datei kann eine beliebige Anzahl an Ausdrücken auf der obersten Ebene beinhalten, wobei ein Ausdruck der obersten Ebene einer der folgenden sein kann:

- Eine Ausgabedefinition, wie \paper, \midi und \layout. Derartige Definitionen auf oberster Ebene verändern die globalen Einstellungen für das ganze „Buch“. Wenn mehr als eine derartige Definition desselben Typs auf oberster Ebene angegeben wird, hat die spätere Vorrang. Für Einzelheiten, wie dadurch die \layout-Umgebung beeinflusst wird, siehe Abschnitt 4.2.1 [Die layout-Umgebung], Seite 509.
- Eine \header-Umgebung. Damit wird die globale Titelseite eingestellt. Das ist die Umgebung, in der sich Definition für das ganze Buch befinden, wie Komponist, Titel usw.
- Eine \score-Umgebung. Die in ihr enthaltene Partitur wird zusammen mit anderen vorkommenden \score-Umgebungen gesammelt und in ein \book zusammengefasst. Dieses Verhalten kann verändert werden, indem die Variable toplevel-score-handler auf höchster Ebene gesetzt wird. Die Definition des Standards findet sich in der Datei ../scm/lily.scm.
- Eine \book-Umgebung fasst mehrere Sätze (d. h. mehrere \score-Umgebungen) logisch in ein Dokument zusammen. Wenn mehrere \score-Partituren vorkommen, wird für jede \score-Partitur eine eigene Ausgabedatei erstellt, in der alle in der Umgebung enthaltenen Partituren zusammengefasst sind. Der einzige Grund, explizit eine \book-Umgebung zu setzen, ist, wenn mehrere Ausgabedateien aus einer einzigen Quelldatei erstellt werden sollen. Eine Ausnahme sind lilypond-book-Dokumente, in denen eine \book-Umgebung explizit hinzugefügt werden muss, wenn mehr als eine \score- oder \markup-Umgebung im gleichen Beispiel angezeigt werden soll. Dieses Verhalten kann verändert werden, indem die Variable toplevel-book-handler auf höchster Ebene gesetzt wird. Die Definition des Standards findet sich in der Datei ../scm/lily.scm.
Ein zusammengesetzter musikalischer Ausdruck wie etwa
\{ c'4 d' e'2 \}
Dieses Beispiel wird von LilyPond automatisch in einer \score-Umgebung in einem Buch interpretiert und mit anderen \score-Umgebungen und musikalischen Ausdrücken auf der höchsten Ebene zusammen ausgegeben. Anders gesagt: eine Datei, die nur das obige Beispiel beinhaltet, wird übersetzt zu
\book {
  \score {
    \new Staff {
      \new Voice {
        \{ c'4 d' e'2 \}
      }
    }
    \layout { }
  }
  \paper { }
  \header { }
}
Dieses Verhalten kann verändert werden, indem die Variable toplevel-music-handler auf der obersten Ebene gesetzt wird. Die Definition des Standards findet sich in der Datei ../scm/lily.scm.

Eine Textbeschriftung, eine Strophe etwa:
\markup {
  2. Die erste Zeile der zweiten Strophe.
}
Textbeschriftungen werden über, zwischen oder unter musikalischen Ausdrücken gesetzt, so wie sie notiert werde.

Eine Variable, wie
\footnote { c4 d e d }
Sie kann dann später in der Datei eingesetzt werden, indem \footnote geschrieben wird. Die Bezeichnung der Variable darf nur aus alphabetischen Zeichen bestehen, keine Zahlen, Unter- oder Bindestriche.

Das folgende Beispiel zeigt drei Dinge, die auf der obersten Ebene notiert werden können:
\layout {
  % Zeilen rechtsbündig setzen
  ragged-right = ##t
}
\header {
  title = "Do-re-mi"
}
\{ c'4 d' e2 \}
An einer beliebigen Stelle der Datei kann jede der folgenden lexikalischen Anweisungen notiert werden:
\version
\include
\sourcefilename
• \sourcefileline
• Ein einzeiliger Kommentar, beginnend mit \%.
• Ein mehrzeiliger Kommentar, umgeben von \%( .. \%).

Leerzeichen zwischen Einheiten in der Eingabe werden generell ignoriert und können nach Belieben weggelassen werden oder hinzugefügt werden, um die Lesbarkeit des Codes zu verbessern. Mindestens ein Leerzeichen sollte jedoch unter folgenden Umständen immer eingesetzt werden, um Fehler zu vermeiden:
• Vor und hinter jeder schließenden oder öffnenden Klammer,
• nach jedem Befehl oder jeder Variable, also jeder Einheit, die mit \ beginnt,
• nach jeder Einheit, die als Scheme-Ausdruck interpretiert werden, also alle Einheiten, die mit \# beginnen.
• Alle Einheiten von Scheme-Ausdrücken müssen mit Leerzeichen getrennt werden,
• in Gesangstextabschnitten (lyricmode) müssen Leerzeichen zwischen alle Ausdrücke in \override- und \set-Befehlen gesetzt werden. Insbesondere müssen um Punkte und Gleichheitszeichen in Befehlen wie \override Score.LyricTex.font-size = \#5) und vor dem gesamten Befehl geschrieben werden.

Siehe auch

Handbuch zum Lernen: Abschnitt “Wie eine LilyPond-Eingabe-Datei funktioniert” in Handbuch zum Lernen.

Notationsreferenz: Abschnitt 4.2.1 [Die layout-Umgebung], Seite 509.

3.2 Titel
Fast alle gedruckten Noten beinhalten einen Titel und den Namen des Komponisten, teilweise wird auch noch sehr viel mehr Information zur Verfügung gestellt.

3.2.1 Titel, Kopf- und Fußzeilen erstellen

Wie funktioniert die Titel-Umgebung?
Es gibt zwei Arten von Titelumgebungen: die Hauptumgebung, die über der ersten \score-Umgebung innerhalb eines „book“ notiert wird, und individuelle Titelumgebungen, die innerhalb von \score auftreten können. Textfelder für beide Typen werden in der \header-Umgebung eingegeben.

Wenn in dem „book“ nur eine einzelne Partitur vorkommt, kann die \header-Umgebung innerhalb oder außerhalb der \score-Umgebung geschrieben werden.

\header {
  title = "SUITE I."
  composer = "J. S. Bach."
}

\score {
  \new Staff \relative {
    \clef bass
SUITE I.

J. S. Bach.

Prélude.

Allemande.

Textfelder der Haupttitelumgebung können in allen \score-Umgebungen gesetzt oder manuell unterdrückt werden:

\book {
  \paper {
    print-all-headers = ##t
  }
  \header {
    title = "DAS WOHLTEMPERIRTE CLAVIER"
    subtitle = "TEIL I"
    % Do not display the tagline for this book
    tagline = ##f
  }
}
Kapitel 3: Allgemeine Eingabe und Ausgabe

\markup { \vspace #1 }
\score {
  \new PianoStaff <<
    \new Staff { s1 }
    \new Staff { \clef "bass" s1 }
  >>
  \header {
    \title "PRAELUDIUM I"
    \opus "BWV 846"
    % Do not display the subtitle for this score
    \subtitle ##f
  }
}

\score {
  \new PianoStaff <<
    \new Staff { s1 }
    \new Staff { \clef "bass" s1 }
  >>
  \header {
    \title "FUGA I"
    \subtitle "A 4 VOCI"
    \opus "BWV 846"
    % Do not display the subtitle for this score
    \subtitle ##f
  }
}

}
Siehe auch

Notationsreferenz: Abschnitt 3.1.5 [Die Dateistruktur], Seite 451, [Angepasstes Layout für Titelumgebungen], Seite 461.

Standardlayout von book- und Partitur-Titelumgebungen

Dieses Beispiel zeigt alle \header-(Überschriften)-Variablen:

```latex
\book {
  \header {
    dedication = "Dedication"
    title = "Title"
    subtitle = "Subtitle"
    subsubtitle = "Subsubtitle"
    % The following fields are evenly spread on one line
    % the field "instrument" also appears on following pages
    instrument = \markup \with-color #green "Instrument"
    poet = "Poet"
    composer = "Composer"
    % The following fields are placed at opposite ends of the same line
    meter = "Meter"
    arranger = "Arranger"
    % The following fields are centered at the bottom
    tagline = "tagline goes at the bottom of the last page"
    copyright = "copyright goes at the bottom of the first page"
  }
  \score {
```
Dedication

Title
Subtitle
Subsubtitle

Poet Instrument Composer
Meter Arranger
Piece 1 Opus 1
\timesig.C44/\clef.G

Piece 2 on the same page Opus 2
\timesig.C44/\clef.G

copyright goes at the bottom of the first page
Kapitel 3: Allgemeine Eingabe und Ausgabe

2 Instrument

Beachten Sie:

• Die Instrumentenbezeichnung wird auf jeder Seite wiederholt.

• Nur piece (Stück) und opus werden für eine Partitur (\score) gesetzt, wenn die \paper-Variable print-all-headers auf ##f gesetzt ist (Standard-Einstellung).

• Textfelder, die in einer \header-Umgebung nicht benutzt werden, werden durch \null-Textbeschriftung ersetzt, sodass sie keinen leeren Platz belegen.

• Die Standard-Einstellungen von scoreTitleMarkup platzieren die Felder piece (Stück) und opus zu den gegenüberliegenden Seiten der selben Zeile.

Um die Standard-Einstellungen des Layouts zu ändern, siehe [Angepasstes Layout für Titelumgebungen], Seite 461.

Mit der Variable breakbefore innerhalb einer \header-Umgebung, die für sich auch eine \score-Umgebung darstellt, kann man die Hauptüberschriften auf der ersten Seite allein ausgeben, sodass die Noten (in der score-Umgebung definiert) erst auf der folgenden Seite beginnen.

\book
\header {
  title = "This is my Title"
  subtitle = "This is my Subtitle"
  copyright = "This is the bottom of the first page"
}
\score {
  \repeat unfold 4 { e'' e'' e'' e'' }
  \header {
    piece = "This is the Music"
    breakbefore = ##t
  }
}
This is the bottom of the first page

2
This is the Music

3

Music engraving by LilyPond 2.19.84—www.lilypond.org

Siehe auch
Handbuch zum Lernen: Abschnitt “Wie eine LilyPond-Eingabe-Datei funktioniert” in Handbuch zum Lernen,

Notationsreferenz: Abschnitt 3.1.5 [Die Dateistruktur], Seite 451, [Angepasstes Layout für Titelumgebungen], Seite 461.


Standardlayout von Kopf- und Fußzeilen

Kopf- und Fußzeilen sind Textzeilen, die ganz oben und ganz unten auf der Seite stehen, unabhängig vom Textbereich eines Buches. Sie können mit folgenden \paper-Variablen kontrolliert werden:

- oddHeaderMarkup
- evenHeaderMarkup
- oddFooterMarkup
- evenFooterMarkup

Diese Beschriftungsvariablen können nur auf Textfelder einer Haupttitelumgebung (eine \header-Umgebung auf höchster Ebene, die sich auf alle \score-Umgebungen einer Datei bezieht) zugreifen und sind definiert in der Datei ly/titling-init.ly. In den Standardeinstellungen

- werden Seitenzahlen automatisch ganz oben links (wenn gerade) oder ganz oben rechts (wenn ungerade) gesetzt, beginnend mit der zweiten Seite.
- wird das instrument-Textfeld auf jeder Seite zentriert, beginnend mit der zweiten Seite.
- wird der copyright-Text unten auf der ersten Zeile zentriert.
- wird der Inhalt von tagline unten auf der letzten Seite zentriert und unterhalb des copyright-Texts, wenn es sich nur um eine Seite handelt.
Die Standardeinstellung von \texttt{tagline} kann verändert werden, indem man ein \texttt{tagline}-Feld in die \header-Umgebung auf höchster Ebene schreibt.

\begin{verbatim}
\book {
  \header {
    tagline = "... music notation for Everyone"
  }
  \relative {
    c'4 d e f
  }
}
\end{verbatim}

... music notation for Everyone

Um die \texttt{tagline} ganz zu entfernen, wird ihr Inhalt als \texttt{##f} (falsch) definiert.

3.2.2 Eigene Kopf- und Fußzeilen sowie Titel

Angepasste Textformatierung für Titelumgebungen

Standard-\texttt{\markup}-Befehle können eingesetzt werden, um jeglichen Text in Titeln, Kopf- und Fußzeilen innerhalb der \header-Umgebung zu verändern.

\begin{verbatim}
\score {
  \relative { s1 }
  \header {
    piece = \markup { \fontsize #4 \bold "PRAELUDIUM I" }
    subtitle = \markup { \italic "(Excerpt)" }
  }
}
\end{verbatim}

PRAELUDIUM I
Siehe auch

Notationsreferenz: Abschnitt 1.8.2 [Text formatieren], Seite 223.

Angepasstes Layout für Titelumgebungen
\markup-Befehle in der \header-Umgebung sind sinnvoll für einfaches Formatieren von Text, aber sie gewähren keine genaue Kontrolle über die Positionierung von Titeln. Um die Positionierung von Titeltextfeldern vorzunehmen, eignen sich beide oder eine von folgenden Variablen (die man in der \paper-Umgebung einsetzen muss):

- bookTitleMarkup
- scoreTitleMarkup


Die Standarddefinitionen von scoreTitleMarkup, wie in der Datei ly/titling-init.ly definiert, sind:

\begin{verbatim}
scoreTitleMarkup = \markup { \column { \on-the-fly #print-all-headers { \bookTitleMarkup \hspace #1 } \fill-line { \fromproperty #'header:piece \fromproperty #'header:opus } } }
\end{verbatim}

Dadurch werden die Textfelder piece (Stück) und opus an den gegenüberliegenden Enden der gleichen Zeile platziert:

\begin{verbatim}
\score { { s1 } \header { piece = "PRAELUDIUM I" opus = "BWV 846" } }
\end{verbatim}

PRAELUDIUM I BWV 846

Das folgende Beispiel verändert die Einstellungen von scoreTitleMarkup, sodass das Textfeld piece zentriert wird und in fetter, größerer Schrift erscheint.

\begin{verbatim}
\book { \paper { indent = 0\mm scoreTitleMarkup = \markup { \fill-line { \null } } }
\end{verbatim}
Textfelder, die normalerweise nur im Haupttitel ausgegeben werden, können auch in die Titel einzelner Partituren aufgenommen werden, indem man `print-all-headers` in die `\paper`-Umgebung einfügt. Ein Nachteil dieser Methode ist, dass Textfelder, die tatsächlich nur im Haupttitel erscheinen sollen, manuell für jede Partitur unterdrückt werden müssen. Siehe auch [Wie funktioniert die Titel-Umgebung?], Seite 453.

Um das zu vermeiden, kann das gewünschte Feld zur Definition von `scoreTitleMarkup` hinzugefügt werden. Im folgenden Beispiel wird das Komponistenfeld (`composer`) (normalerweise mit `bookTitleMarkup` assoziiert) zu `scoreTitleMarkup` hinzugefügt, sodass jede Partitur einen eigenen Komponisten haben kann.

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:composer
      }
    }
    \header { tagline = ##f }
  }
  \score {
    { s1 }
    \header {
      piece = "PRAELUDIUM I"
      composer = "Christian Petzold"
    }
  }
}
```

```
PRAELUDIUM I

\begin{music}
\end{music}

```

```
Textfelder, die normalerweise nur im Haupttitel ausgegeben werden, können auch in die Titel einzelner Partituren aufgenommen werden, indem man `print-all-headers` in die `\paper`-Umgebung einfügt. Ein Nachteil dieser Methode ist, dass Textfelder, die tatsächlich nur im Haupttitel erscheinen sollen, manuell für jede Partitur unterdrückt werden müssen. Siehe auch [Wie funktioniert die Titel-Umgebung?], Seite 453.

Um das zu vermeiden, kann das gewünschte Feld zur Definition von `scoreTitleMarkup` hinzugefügt werden. Im folgenden Beispiel wird das Komponistenfeld (`composer`) (normalerweise mit `bookTitleMarkup` assoziiert) zu `scoreTitleMarkup` hinzugefügt, sodass jede Partitur einen eigenen Komponisten haben kann.

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:composer
      }
    }
    \header { tagline = ##f }
  }
  \score {
    { s1 }
    \header {
      piece = "PRAELUDIUM I"
      composer = "Christian Petzold"
    }
  }
}
```
Es ist auch möglich, eigene Textfelder zu erstellen und dann auf sie in der Beschriftungsdefinition zu verweisen:

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \override #`(direction . ,UP) {
          \dir-column {
            \center-align \fontsize #-1 \bold
            \fromproperty #'header:mycustomtext %% User-defined field
            \center-align \fontsize #4 \bold
            \fromproperty #'header:piece
          }
          \fromproperty #'header:opus
        }
      }
    }
    \header { tagline = ##f }
  }
  \score {
    \s1 }
  \header {
    piece = "FUGA I"
    mycustomtext = "A 4 VOCI" %% User-defined field
    opus = "BWV 846"
  }
}
```
Siehe auch

Notationsreferenz: [Wie funktioniert die Titel-Umgebung?], Seite 453.

Angepasstes Layout für Kopf- und Fußzeilen

\markup-Befehle in der \header-Umgebung sind nützlich um einfachen Text zu formatieren, erlauben aber keine Kontrolle über die Positionierung von Kopf- und Fußzeilen. Um die Positionierung der Textfelder zu ändern, eignen sich beide oder eine von folgenden Variablen (die man in der \paper-Umgebung einsetzen muss):

- oddHeaderMarkup
- evenHeaderMarkup
- oddFooterMarkup
- evenFooterMarkup

Der \markup-Befehl \on-the-fly kann eingesetzt werden, um Beschriftung anhand einer Bedingung zu Kopf- und Fußzeilentext innerhalb der \paper-Umgebung hinzuzufügen. Hierzu wird folgende Syntax eingesetzt:

```
Variable = \markup {
  ... 
  \on-the-fly #Prozedur Beschriftung
  ...
}
```

Die Prozedur wird jedes mal aufgerufen, wenn der \markup-Befehl, auf den sie sich auswirkt, ausgewertet wird. Die Prozedur sollte nach einer bestimmten Bedingung fragen und das Beschriftung-Argument ausfüllen (also setzen), wenn diese Bedingung wahr ist.

Eine Anzahl von fertigen Prozeduren, um verschiedene Bedingungen zu testen, werden bereitgestellt:

<table>
<thead>
<tr>
<th>Prozedurbezeichnung</th>
<th>Getestete Bedingung</th>
</tr>
</thead>
<tbody>
<tr>
<td>print-page-number-check-first</td>
<td>sollte diese Seitenzahl gesetzt werden?</td>
</tr>
<tr>
<td>create-page-number-stencil</td>
<td>'print-page-numbers wahr?</td>
</tr>
<tr>
<td>print-all-headers</td>
<td>'print-all-headers wahr?</td>
</tr>
<tr>
<td>first-page</td>
<td>erste Seite im Buch?</td>
</tr>
<tr>
<td>(on-page nmbr)</td>
<td>Seitenzahl = nmbr?</td>
</tr>
<tr>
<td>last-page</td>
<td>letzte Seite im Buch?</td>
</tr>
<tr>
<td>not-first-page</td>
<td>nicht erste Seite im Buch?</td>
</tr>
<tr>
<td>part-first-page</td>
<td>erste Seite im book part?</td>
</tr>
<tr>
<td>part-last-page</td>
<td>letzte Seite im book part?</td>
</tr>
<tr>
<td>not-single-page</td>
<td>Seiten im book part &gt; 1?</td>
</tr>
</tbody>
</table>

Das folgende Beispiel zentriert die Seitenzahlen unten auf jeder Seite. Zuerst werden die Standardeinstellungen von oddHeaderMarkup und evenHeaderMarkup entfernt, indem sie als „Null“-Beschriftung definiert werden. Dann wird oddFooterMarkup mit zentrierter Seitennummer neu definiert. Schließlich erhält evenFooterMarkup die selbe Layoutdefinition wie \oddFooterMarkup:

```
\book {
  
  \markup {
    ...
    \on-the-fly #Prozedur
    ...
  }

  \oddFooterMarkup {
    \on-the-fly #Prozedur
  }

  \evenFooterMarkup {
    \on-the-fly #Prozedur
  }

  \oddHeaderMarkup {
    \on-the-fly #Prozedur
  }

  \evenHeaderMarkup {
    \on-the-fly #Prozedur
  }
```

FUGA I
A 4 VOCI
BWV 846
Verschiedene \on-the-fly-Bedingungen können mit dem Operator „UND“ verknüpft werden, beispielsweise bestimmt

\on-the-fly #first-page
\on-the-fly #last-page
{ \markup ... \fromproperty #'header: ... }

ob es sich um eine einzelne Seite bei der Ausgabe handelt.

Siehe auch


3.2.3 Fußnoten erstellen

Zwei Arten an Fußnoten können erstellt werden: automatische und manuelle Fußnoten.
Übersicht über Fußnoten

Automatische Fußnoten erstellen aufsteigende Zahlenverweise, während mit manuellen Fußnoten eigene angepasste Verweise erstellt werden können. Fußnoten werden normalerweise wie ein \texttt{\tweak}-Befehl an Noten gehängt und können deshalb auch direkt mit den Grobs (graphischen Objekten) verknüpft werden, die von den meisten musikalischen Elementen und Postelementen erstellt werden. In Fällen, wo das nicht funktioniert (etwa bei Taktstrichen und Taktartänderungen, wo die Grobs als Folge einer Eigenschaftsänderung erstellt werden), können Fußnoten als einzelnstehendes musikalisches Ereignis erstellt werden, das sich auf alle Grobs eines bestimmten Typs zu einer bestimmten Zeit bezieht.

Der vollständige Befehl ist:

\footnote \textcolor{white}{zeichen} \textcolor{black}{\textit{Verschiebung}} \textcolor{white}{\textit{Grob-Bezeichnung}} \textcolor{black}{\textit{Fußnote}} \textcolor{white}{\textit{Noten}}

Die Elemente sind folgende:

- \textit{Zeichen} ist eine Beschriftung oder Zeichenkette, die das Fußnotenzeichen angibt, welches für den Referenzpunkt als auch für die Fußnote unten auf der Seite benutzt wird. Es kann ausgelassen werden (oder durch \texttt{\default} ersetzt werden); in diesem Fall wird eine aufsteigende Zahlenfolge erstellt.

- \textit{Verschiebung} (offset) ist ein Zahlenpaar wie etwa \texttt{#(2 . 1)}, das die X- und Y-Verschiebung vom Referenzpunkt aus angibt, wo das Zeichen gesetzt werden soll.

- \textit{Grob-Bezeichnung} gibt die Grob-Art an, der ein Fußnotenzeichen hinzugefügt werden soll (wie etwa \texttt{#'Flag'}). Wenn sie angegeben wird, wird der entsprechende Grob als Referenzpunkt eingesetzt, auch wenn das referenzierte Element nicht die \textit{Noten} selber sind, sondern ein Grob, der durch sie erstellt wird. Das Element kann ausgelassen werden (oder durch \texttt{\default} ersetzt werden); dann wird nur ein direkt erstellter Grob mit Fußnote versehen.

- \textit{Fußnote} Diese Beschriftung oder Zeichenkette bezeichnet den Fußnotentext, der am unteren Seitenrand gesetzt werden soll.

- \textit{Noten} Das ist das Element, ein musikalisches Ereignis oder eine Akkordkonstruktion oder ein Post-Ereignis, das die Fußnote erhält. Man kann es nicht auslassen, aber man kann es durch \texttt{\default} ersetzen. In diesem Fall wird die Fußnote aber nicht an einen bestimmten musikalischen Ausdruck angehängt, sondern an einen zeitlichen Moment. In diesem Fall muss man zwingend die \textit{Grob-Bezeichnung} angeben, um den Grob-Typ auszuwählen, auf den sich die Fußnote bezieht (etwa \texttt{#'TimeSignature'}).

Wie auch mit \texttt{\tweak} muss dem \texttt{\footnote}-Befehl – vorangestellt werde,n wenn er an ein Post-Ereignis oder eine Artikulation angehängt werden soll, damit der Parser das Ergebnis der vorherigen Note oder Pause zuordnen kann.

Automatische Fußnoten

Automatische Fußnoten haben vier Argumente: die \texttt{(x . y)}-Position des Indikators, die optionale \textit{Grob-Bezeichnung}, die die Anmerkung erhalten soll, die \textit{Fußnote}-Beschriftung, die den Inhalt der Fußnote enthält, und natürlich die \textit{Noten}, welche mit einer Fußnote versehen werden sollen.

Der Befehl \texttt{\footnote} muss \texttt{vor} dem Grob geschrieben werden, auf den sich die Fußnote bezieht.

\texttt{\book {
  \header { tagline = ##f } 
  \relative c' { 
}
Die erste Note

Die dritte Note

Noten in Akkorden stellen keine Schwierigkeit dar:

\footnote #'(1 . -1.25) "Hier ein C" c
\footnote #'(2 . -0.25) \markup { \italic "Ein Es" } es
\footnote #'(2 . 3) \markup { \bold "Das ist ein G" } g

\footnote #'(0.5 . -2) \markup { Die erste Note } a'4 b8
\footnote #'(0.5 . 1) \markup { Die dritte Note } Flag e\noBeam c4 d4

\book {
  \header { tagline = ##f }
  \relative c' {
    \footnote #(1 . -1.25) "Hier ein C" c
    \footnote #(2 . -0.25) \markup { \italic "Ein Es" } es
    \footnote #(2 . 3) \markup { \bold "Das ist ein G" } g
  }
}

\footnote #'(0.5 . -2) \markup { Die erste Note }
a'4 b8
\footnote #'(0.5 . 1) \markup { Die dritte Note } Flag e\noBeam c4 d4

\footnote #'(0.5 . 1) \markup { Die dritte Note } Flag e\noBeam c4 d4
Hier noch einige Beispiele von Grobs, die mit Fußnoten versehen sind, wobei auch die Position des Fußnotentextes relativ zu Tagline und Copyright gezeigt wird.

\book {
  \header { copyright = \markup { "Copyright 1970" } }
  \relative {
    a'4\footnote #'(-3 . 0) \markup { \bold Forte } \f
    ~\footnote #'(0 . 1.5) \markup { Ein Bogen } ( b8)\footnote #'(0 . -2) \markup { Balken } [ e]
    \single\footnote #'(1 . -1)
    \markup { \teeny { Das ist ein Hals } } Stem c4
    \single\footnote #'(0 . 0.5)
    \markup \italic { Ein warnendes Versetzungsziehen } AccidentalCautionary
    \footnote #'(1 . 1) "Die Note selber"
    dis?4\footnote #'(0.5 . -0.5) \markup \italic { langsamer werden }
      _"rit."
  }
}

\begin{music}
\[2\]
\end{music}

1Ein Bogen
2Forte
3Balken
4Das ist ein Hals
5Die Note selber
6Ein warnendes Versetzungsziehen
7langsamer werden

Copyright 1970

Music engraving by LilyPond 2.19.84—www.lilypond.org

Für eine \markup-Beschriftung auf oberster Ebene braucht man den Befehl \auto-footnote:

\book {
  \header { tagline = ##f }
  \markup { \auto-footnote "Eine einfache Melodie" \italic "Von mir" }
  \relative {
    a'4 b8 e c4 d
  }
}

\footnote 1
\footnote 2
\footnote 3
\footnote 4
\footnote 5
\footnote 6
\footnote 7
Eine einfache Melodie\footnote{Von mir}

Manuelle Fußnoten

Manuell erstellte Fußnoten haben ein zusätzliches erstes Argument \textit{Zeichen}, das das Fußnotenzeichen erstellt. Im Gegensatz zu automatisch erstellten Fußnotenzeichen erscheinen sie nicht unbedingt vor dem Fußnotentext unten auf der Seite – das Herstellen eines visuellen Zusammenhanges ist dem Setzer überlassen. LilyPond stellt nur sicher, dass der zugehörige Text unten auf der selben Seite erscheint.

Davon abgesehen ist die Benutzung identisch mit automatischen Fußnoten.
Kapitel 3: Allgemeine Eingabe und Ausgabe

1. Die erste Note
2. Die zweite Note
3. Piano

Fußnoten für Akkordnoten werden wie folgt notiert:

\book {
  \header { tagline = ##f }
  \relative c' {
    <
    \footnote "1" #'(1 . -1.25) "1. C" c
    \footnote "b" #'(2 . -0.25) "b. E-flat" es
    \footnote "3" #'(2 . 3) \markup { \italic "iii. G" } g
  }
}

\footnote "1" #'(1 . -1.25) "1. C" c
\footnote "b" #'(2 . -0.25) "b. E-flat" es
\footnote "3" #'(2 . 3) \markup { \italic "iii. G" } g

Achtung: Wenn Fußnoten die selbe vertikale Position haben, werden sie von oben nach unten nummeriert.

Hier einige Beispiele manueller Fußnoten, die auch die relative Position der Fußnotentexte zu Tagline und Copyright anzeigen:

\book {
  \header { tagline = ##f }
  \relative {

iii. G
b. E-flat
1. C
Eine \markup-Beschriftung der höchsten Ebene wird wie folgt annotiert:

\book{
  \header{tagline=##f}
  \markup{"Eine einfache Melodie" \footnote{*} \italic{* Von mir}}
  \relative{
    \space{2}
a'4 b8 e c4 d4
  }
Eine einfache Melodie *

* Von mir

Siehe auch


Notationsreferenz: [Erklärungen in Ballonform], Seite 211, Abschnitt 4.1 [Seitenlayout], Seite 498, [Textartige Zeichen], Seite 219, [Textarten], Seite 216, Abschnitt 3.2 [Titel], Seite 453.


Bekannte Probleme und Warnungen

Mehrere Fußnoten auf einer Seite können ausschließlich übereinander gedruckt werden und nicht auf der gleichen Zeile gesetzt werden. Fußnoten können nicht an MultiMeasureRests (mehrtaktige Pause) angehängt werden und können mit Staff-(Notensystem)-, \markup-(Beschriftungs)-Objekten und anderen Fußnoten zusammenstoßen. Wenn man den manuellen Fußnotenbefehl einsetzt, braucht man in der \paper-Umgebung den Eintrag footnote-auto-number = ##f.

3.2.4 Verweis auf die Seitenzahlen

Eine bestimmte Stelle der Partitur kann mit einem \label-Befehl markiert werden, sowohl auf oberster Ebene als auch innerhalb eines musikalischen Ausdrucks. Auf diese Marke kann dann verwiesen werden, um die Seitenzahl zu erhalten, auf der die Marke vorkommt. Der Verweis wird mit dem Befehl \page-ref gefordert (innerhalb von \markup).

\header { tagline = ##f }\book {\label #'ErstePartitur\score { { c'1 \pageBreak \mark A \label #'ZeichenA c'1 } } }\markup { Die erste Partitur fängt auf Seite \page-ref #'ErstePartitur "0" "?" an.}
Die erste Partitur fängt auf Seite 1 an.
Zeichen A befindet sich auf Seite 2.
Der \page-ref-Textbeschriftungsbefehl braucht drei Argumente:
1. die Marke, ein Scheme-Symbol, etwa #'ErstePartitur,
2. eine Beschriftung, die als Platzhalter benutzt wird, um die Breite des Verweisen zu schätzen,
3. eine Beschriftung, die anstelle der Seitenzahl gesetzt wird, wenn die Marke unbekannt ist.

Der Grund, warum ein Platzhalter benötigt wird, ist dass zu dem Zeitpunkt, an dem die Textbeschriftungen ausgewertet werden, noch keine Seitenumbrüche vorgenommen wurden und die Seitenzahlen deshalb noch nicht bekannt sind. Um hier ein Problem zu vermeiden, wird die eigentliche Auswertung der Textbeschriftung erst später ausgeführt, die Größe des Textes muss aber schon vorher bekannt sein. Die Größe wird mithilfe des Platzhalters bestimmt. Wenn eine Partitur zwischen 10 und 99 Seiten hat, kann man "00" schreiben, also eine zweistellige Zahl.

Vordefinierte Befehle

3.2.5 Inhaltsverzeichnis

Ein Inhaltsverzeichnis kann eingefügt werden mit dem Befehl \markuplist \table-of-contents. Die Elemente, die im Inhaltsverzeichnis aufgelistet werden sollen, werden mit dem \tocItem-Befehl markiert, welches sowohl auf höchster Ebene als auch in einem musikalischen Ausdruck verwendet werden kann.

\markuplist \table-of-contents
\pageBreak

\tocItem \markup "Erste Partitur"
\score {
  
  c'4 % ...
  \tocItem \markup "Ein bestimmter Punkt innerhalb der ersten Partitur"
  d'4 % ...
}

\tocItem \markup "zweite Partitur"
\score {
  
  e'4 % ...
}

Die Beschriftungen, die benutzt werden um das Inhaltsverzeichnis zu formatieren, sind in der \paper-Umgebung definiert. Die Standardformatierungselemente sind tocTitleMarkup um die Überschrift zu formatieren und tocItemMarkup um die einzelnen Inhaltslemente zu formatieren, bestehend aus dem Titelelement und einer Seitenzahl. Die Variablen können durch den Benutzer geändert werden:

\paper {
  \% Übersetzung der Inhaltsverzeichnisüberschrift nach französisch:
  tocTitleMarkup = \markup \huge \column {
    \fill-line { \null "Table des matières" \null }
    \hspace #1
  }
  \% hier größere Schriftarten
  tocItemMarkup = \markup \large \fill-line {
}

Neue Befehle und Beschriftungen können auch definiert werden, um eigene Inhaltsverzeichnisse zu gestalten:

- zuerst muss eine neue Beschriftungsvariable in der \paper-Umgebung definiert werden
- dann muss die musikalische Funktion definiert werden, die ein Element zum Inhaltsverzeichnis hinzufügt, indem die neue Variable benutzt wird.

Das folgende Beispiel definiert einen neuen Stil um Akt-Bezeichnungen einer Oper in das Inhaltsverzeichnis aufzunehmen:

\paper {
  tocActMarkup = \markup \large \column {
    \hspace #1
    \fill-line { \null \italic \fromproperty #'toc:text \null }
    \hspace #1
  }
}

tocAct =
#(define-music-function (text) (markup?)
  (add-toc-item! 'tocActMarkup text))

Table of Contents

Atto Primo

Coro. Viva il nostro Alcide 1
Cesare. Presti omai l'Egizzia terra 1

Atto Secondo

Sinfonia 1
Cleopatra. V'adoro, pupille, saette d'Amore 1

Die Zeile zwischen dem Inhalt und der Seitenzahl kann mit einer gepunkteten Linie gefüllt werden:

\header { tagline = ##f }
\paper {
  tocItemMarkup = \tocItemWithDotsMarkup
}

\book {
  \markuptext \table-of-contents
  \tocItem \markup { Allegro }
  \tocItem \markup { Largo }
  \markup \null
}
Table of Contents

Allegro ........................................ 1
Largo ........................................... 1

Siehe auch

Installierte Dateien: ly/toc-init.ly.

Vordefinierte Befehle

\table-of-contents, \tocItem.

3.3 Arbeiten an Eingabe-Dateien

3.3.1 LilyPond-Dateien einfügen

Ein größeres Projekt kann in einzelne Dateien aufgeteilt werden. Um eine andere Datei einzubinden, kann der Befehl

\include "andereDatei.ly"

benutzt werden.

Die Zeile \include "andereDatei.ly" benimmt sich genauso, also ob der Inhalt der Datei andereDatei.ly komplett in die Hauptdatei eingefügt werden würde. So kann man für ein größeres Projekt die einzelnen Stimmen der Instrumente getrennt notieren und sie dann in einer Partitur-Datei benutzen. Meistens werden in den eingefügten Dateien einige Variablen definiert, die dann auch in der Hauptdatei eingesetzt werden können. Mit Marken (Tags) gekennzeichnete Abschnitte können eingesetzt werden, um die entsprechenden Noten etc. an verschiedenen Stellen in der Datei zur Verfügung zu stellen. Siehe auch Abschnitt 3.3.2 [Verschiedene Editionen aus einer Quelldatei], Seite 477.

Auf Dateien im aktuellen Verzeichnis kann einfach mit dem Dateinamen nach dem \include-Befehl verwiesen werden. Dateien an anderen Stellen können eingebunden werden, indem entweder ein vollständiger Pfad oder ein relater Pfad zu der Datei angegeben wird. Hierbei sollten die für UNIX typischen Schrägstriche (/) und nicht die rückwärtsgeneigten von Windows (\) verwendet werden, um die Verzeichnisse zu trennen. Wenn etwa die Datei kram.ly ein Verzeichnis höher liegt als das aktuelle Verzeichnis, sollte der Befehl so aussehen:

\include "../kram.ly"

Wenn die Orchesterstimmen andererseits in einem Unterordner mit der Bezeichnung stimmen liegen, sieht er folgendermaßen aus:

\include "stimmen/VI.ly"
\include "stimmen/VII.ly"
... etc

Dateien, die eingebunden werden sollen, können selber auch wiederum ein \include enthalten. Diese Einbindung zweiter Ebene werden erst interpretiert, wenn sie sich in der Hauptdatei befinden, sodass die Pfadangaben hier nicht relativ zur eingebundenen Datei, sondern relativ zur Hauptdatei gesetzt werden müssen. Dieses Verhalten kann jedoch auch verändert werden, indem man lilypond die Option -drelative-includes auf der Komandozeile zuweist (oder indem man den Befehl #(ly:set-option 'relative-includes #t) an den Beginn der Quelldatei schreibt). Mit relative-includes wird der Pfad jedes \include-Befehls als relativ zu
Kapitel 3: Allgemeine Eingabe und Ausgabe

... der Datei angenommen, in der sich der Befehl befindet. Dieses Verhalten wird empfohlen und wird in zukünftigen Versionen von LilyPond den Standard darstellen.

Dateien können auch aus einem Verzeichnis eingebunden werden, dass im Suchpfad von LilyPond liegt. Hierzu muss auf der Kommandozeile das entsprechende Verzeichnis angegeben werden und die Dateien, die eingebunden werden, müssen nur mit ihrem Namen notiert sein. Wenn etwa die Datei Haupt.ly kompiliert werden soll, die Dateien aus dem Unterverzeichnis stimmen einbindet, müssen sie sich im Verzeichnis von Haupt.ly befinden und dann LilyPond folgendermaßen aufrufen:

```plaintext
lilypond --include=stimmen Haupt.ly
```

In Haupt.ly steht:

```plaintext
\include "VI.ly"
\include "VII.ly"
... usw.
```

Dateien, die in vielen Partituren verwendet werden sollen, können im LilyPond-Verzeichnis ./.ly gespeichert werden. (Die Stelle, an der dieses Verzeichnis sich befindet, hängt vom Betriebssystem ab, siehe hierzu Abschnitt “Mehr Information” in Handbuch zum Lernen). Dateien in diesem Verzeichnis können einfach mit ihrem Namen eingefügt werden. So werden auch die Sprachdateien wie etwa deutsch.ly eingefügt.


Einige einfache Beispiele, die die Benutzung von `\include` demonstrieren, sind dargestellt in Abschnitt “Partituren und Stimmen” in Handbuch zum Lernen.

Siehe auch


Bekannte Probleme und Warnungen

Wenn eine Datei eingebunden wird, deren Name einer Datei aus dem Installationsverzeichnis von LilyPond entspricht, wird die installierte Datei anstelle der eigenen verwendet.

3.3.2 Verschiedene Editionen aus einer Quelldatei

Es gibt verschiedene Funktionen, die es möglich machen, unterschiedliche Versionen einer Partitur aus der gleichen Quelldatei zu produzieren. Variablen werden am besten eingesetzt, wenn es darum geht, längere Notenpassagen und/oder Anmerkungen/Textmarken miteinander auf verschiedene Weise zu kombinieren. Tag-Marken dagegen werden am besten eingesetzt, wenn eine von mehreren kurzen alternativen Notenabschnitten ausgewählt werden soll und können auch eingesetzt werden, um Musikabschnitte an unterschiedlichen Stellen zusammenzufügen.

Egal welche Methode am Ende eingesetzt wird: Es erleichtert die Arbeit in jedem Fall, wenn die eigentlichen Noten und die Struktur der Partitur voneinander getrennt notiert werden – so kann die Struktur geändert werden, ohne dass man Änderungen an den Noten vornehmen muss.

Variablen benutzen

Wenn Notenabschnitt in Variablen definiert werden, können sie an unterschiedlichen Stellen in der Partitur eingesetzt werden, siehe auch Abschnitt “Stücke durch Variablen organisieren” in
Handbuch zum Lernen. Zum Beispiel enthält eine Vokalpartitur für ein *a cappella* Stück oft einen Klavierauszug, der das Einführen einfacher macht. Der Klavierauszug enthält die gleichen Noten, sodass man sie nur einmal notieren muss. Noten aus zwei Variablen können auf einem System kombiniert werden, siehe [Automatische Kombination von Stimmen], Seite 166. Hier ein Beispiel:

```latex
sopranoMusic = \relative { a'4 b c b8( a) }
altoMusic = \relative { e'4 e e f } 
tenorMusic = \relative { c'4 b e d8( c) }
bassMusic = \relative { a4 gis a d, }
allLyrics = \lyricmode {King of glory -- ry }
<<
\new Staff = "Soprano" \sopranoMusic
\new Lyrics \allLyrics
\new Staff = "Alto" \altoMusic
\new Lyrics \allLyrics
\new Staff = "Tenor" {
    \clef "treble_8"
    \tenorMusic
}
\new Lyrics \allLyrics
\new Staff = "Bass" {
    \clef "bass"
    \bassMusic
}
\new Lyrics \allLyrics
\new PianoStaff <<
\new Staff = "RH" {
    \set Staff.printPartCombineTexts = ##f
    \partcombine
    \sopranoMusic
    \altoMusic
}
\new Staff = "LH" {
    \set Staff.printPartCombineTexts = ##f
    \clef "bass"
    \partcombine
    \tenorMusic
    \bassMusic
}
>>
>>
```
Unterschiedliche Partituren, die entweder nur den Chor oder das Klavier zeigen, können produziert werden, indem die Struktur verändert wird; die Noten müssen dazu nicht verändert werden.

Für längere Partituren können Variablen in eigene Dateien notiert werden, die dann einge- bunden werden, siehe Abschnitt 3.3.1 [LilyPond-Dateien einfügen], Seite 476.

**Marken benutzen**

Der \tag \#‘TeilA-Befehl markiert einen musikalischen Ausdruck mit der Bezeichnung TeilA. Ausdrücke, die auf diese Weise markiert werden, können mit ihrer Bezeichnung später ausgewählt bzw. ausgefiltert werden. Das geschieht mit den Befehlen \keepWithTag \#‘Bezeichnung bzw. \removeWithTag \#‘Bezeichnung. Die Wirkung dieser Filter auf die markierten Notenabschnitte ist wie folgt:

<table>
<thead>
<tr>
<th>Filter</th>
<th>Resultat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markierte Noten mit vorgesetztem \keepWithTag #‘Bezeichnung</td>
<td>Unmarkierte Noten und Noten mit der Marke Bezeichnung werden gesetzt, Noten mit einer anderen Marke werden nicht angezeigt.</td>
</tr>
<tr>
<td>Markierte Noten mit vorgesetztem \removeWithTag #‘Bezeichnung</td>
<td>Unmarkierte Noten und Noten mit einer anderen Marke als Bezeichnung wird angezeigt, Noten markiert mit Bezeichnung werden nicht angezeigt.</td>
</tr>
<tr>
<td>Markierte Noten, weder mit vorgesetztem \keepWithTag noch \removeWithTag</td>
<td>Alle markierten und unmarkierten Noten werden angezeigt.</td>
</tr>
</tbody>
</table>

Die Argumente der Befehle \tag, \keepWithTag und \removeWithTag sollten ein Symbol sein (wie etwa \#‘score oder \#‘part), gefolgt von einem musikalischen Ausdruck.

Im folgenden Beispiel erscheinen zwei Versionen der Noten, eine zeigt Triller in normaler Notation, die andere zeigt sie ausgeschrieben:
Kapitel 3: Allgemeine Eingabe und Ausgabe

Entsprechend können auch Abschnitte ausgeschlossen werden; das erfordert manchmal weniger Schreibarbeit:

\begin{music}
\relative { 
g'8. c32 d 
\tag #'trills { d8.\trill } 
\tag #'expand { \repeat unfold 3 { e32 d } } 
c32 d 
}
\end{music}

\begin{music}
\relative { 
g'8. c32 d 
\tag #'trills { d8.\trill } 
\tag #'expand { \repeat unfold 3 { e32 d } } 
c32 d 
}
\end{music}

Marken können auch auf Artikulationen, Text usw angewendet werden, indem man ihnen
- \tag #'your-tag
Kapitel 3: Allgemeine Eingabe und Ausgabe

voranstellt (jedoch nach der Note, an die sie gebunden sind). Mit diesem Code etwa könnte man entweder Fingersatz oder aber einen Text ausgeben:

\begin{verbatim}
c1-\tag #'finger ^4
c1-\tag #'warn ^"Achtung!"
\end{verbatim}

Mehrfache Marken können mithilfe von mehreren \tag-Befehlen notiert werden:

\begin{verbatim}
music = \relative c'' {
   \tag #'a \tag #'both { a4 a a a }
   \tag #'b \tag #'both { b4 b b b }
}
<<
\keepWithTag #'a \music
\keepWithTag #'b \music
\keepWithTag #'both \music
>>
\end{verbatim}

Mehrfache \removeWithTag-Filter können auf einen musikalischen Ausdruck angewendet werden, um mehrere unterschiedliche markierte Abschnitte aus dem Druckbild zu entfernen.

\begin{verbatim}
music = \relative c'' {
   \tag #'A { a4 a a a }
   \tag #'B { b4 b b b }
   \tag #'C { c4 c c c }
   \tag #'D { d4 d d d }
}
{\removeWithTag #'B
 \removeWithTag #'C
 \music}
\end{verbatim}

Zwei oder mehr \keepWithTag-Filter in einem musikalischen Ausdruck bewirken, dass alle markierten Abschnitte entfernt werden, weil der erste Befehl alle markierten Abschnitt außer dem im Befehl genannten wegfiltiert und der zweite Befehl dann auch diesen eben genannten zusätzlich entfernt.

Manchmal will man Noten an einem bestimmten Platz in existierenden Noten einfügen. Dafür kann entweder \pushToTag oder \appendToTag benutzt werden, um Material hinter bzw. vor den
Elementen (elements) der existierenden Noten einzufügen. Nicht alle musikalischen Konstruktionen haben Elemente, aber sequentielle und simultane Noten sind ziemlich gute Kandidaten:

\[
\text{test} = \{ \text{\tag #'here} \{ \text{\tag #'here} <<c''>> \} \}
\]

\{
\pushToTag #'here c'
\pushToTag #'here e'
\pushToTag #'here g' \text{\test}
\appendToTag #'here c'
\appendToTag #'here e'
\appendToTag #'here g' \text{\test}
\}

Beide Befehle erhalten einen Tag, das Material, das bei jedem Auftreten des Tags eingefügt werden soll und den Ausdruck, der mit dem Tag versehen ist. Der Befehl stellt sicher, dass alle Anderungen kopiert werden, sodass das ursprüngliche \text{test} seine Bedeutung behält.

Siehe auch


Notationsreferenz: [Automatische Kombination von Stimmen], Seite 166, Abschnitt 3.3.1 [LilyPond-Dateien einfügen], Seite 476.

Globale Einstellungen benutzen

Man kann globale Einstellungen aus einer externen Datei einfügen:

\text{lilypond -dinclude-settings=MY_SETTINGS.ly MY_SCORE.ly}


Diese Technik funktioniert auch gut für Formatvorlagen, wie in Abschnitt “Formatvorlagen” in Handbuch zum Lernen behandelt.

Siehe auch


Notationsreferenz: Abschnitt 3.3.1 [LilyPond-Dateien einfügen], Seite 476.

3.3.3 sonderzeichen

Zeichenkodierung

Das Aussehen des Zeichens wird bestimmt durch die gerade benutzte Schriftart (engl. font). In einer Schriftartdatei werden die Nummern der Unicode-Zeichen einem bestimmten Glyphen zugeordnet. LilyPond verwendet die Pango-Bibliothek um mehrsprachige Texte und komplexe Skripte korrekt zu setzen.


Wenn eine LilyPond-Eingabedatei nicht-ASCII-Zeichen enthält und nicht in UTF-8 gespeichert ist, gibt es folgende Fehlermeldung:

```
FT_Get_Glyph_Name () error: invalid argument
```

Hier ein Beispiel mit Kyrilliza, hebräischem und portugiesischem Text:

![Kyrilliza, hebräischem und portugiesischem Text](noteheads.s1)

### Unicode

Um einen einzelnen Buchstaben zu notieren, für den die Unicode-Ziffernfolge bekannt ist, der aber nicht auf der Tastatur zu finden ist, kann der Befehl `\char ##xhhhh` oder `\char #dddd` innerhalb einer `\markup`-Umgebung benutzt werden. Hierbei bedeutet `hhhh` die hexadezimale Zahl und `ddd` die entsprechende dezimale Zahl für das erforderliche Zeichen. Nullen zu Beginn können ausgelassen werden, aber normalerweise werden alle vier Zeichen der hexadezimalen Notation notiert. (Achten Sie darauf, dass Sie nicht UTF-8-Codepunkte einsetzen, weil UTF-8 zusätzliche Bits enthält, die die Nummer der Oktets bezeichnet.) Unicode-Tabellen und ein Verzeichnis der Zeichenbezeichnungen mit einer hexadezimalen Verweiszahl finden sich auf der Internetseite des Unicode Consortiums: [http://www.unicode.org/](http://www.unicode.org/).

Mit `\char ##x03BE` und `\char #958` wird beispielsweise das Unicode-Zeichen U+03BE notiert, welches die Unicode-Bezeichnung „Greek Small Letter Xi“ hat.

Alle existierenden Unicode-Zeichen können auf diese Weise notiert werden, und wenn für alle Zeichen dieses Format angewendet wird, muss die Datei nicht im utf-8-Format gespeichert werden. Es muss natürlich auch noch eine Schriftart auf dem System installiert sein, die die notierten Zeichen darstellen kann.


```
\score { 
  \relative { 
    c''1 \mark \markup { \char ##x03EE } 
    c1. \markup { \tiny { \char ##x03B1 " to " \char ##x03C9 } } 
  } 
  \addlyrics { 0 \markup { \concat { Ph \char ##x0153 be! } } } 
} 
\markup { "Copyright 2008--2015" \char #x00A9 }
```
Copyright 2008–2015 ©

Um das Copyright-Zeichen zu notieren, kann folgender Code eingesetzt werden:

\header {
  copyright = \markup { \char ##x00A9 "2008" }
}

ASCII-Aliase

Eine Liste von ASCII-Befehlen für Sonderzeichen kann eingefügt werden:

\paper {
  #(include-special-characters)
}

\markup "&flqq; &ndash; &Œ;uvre incomplète&hellip; &frqq;"

\score {
  \new Staff { \repeat unfold 9 a'4 }
  \addlyrics {
    This is al -- so wor -- kin'~in ly -- rics: &ndash;,&Œ;&hellip;
  }
}

\markup \column {
  "The replacement can be disabled:"
  "&ndash;,&Œ; &hellip;"
  \override #'(replacement-alist . ()) "&ndash;,&Œ; &hellip;"
}

« – Œuvre incomplète... »

This is al-so workin'in lyrics: – Œ...

The replacement can be disabled:
– Œ... &ndash; &Œ; &hellip;

Man kann auch eigen Aliase erstellen, entweder global:

\paper {
  #(add-text-replacements!
    '(("100" . "hundred")
      ("dpi" . "dots per inch")))
}
\markup "A 100 dpi."
A hundred dots per inch.

oder lokal:

\markup \replace #'(("100" . "hundred")
                 ("dpi" . "dots per inch")) "A 100 dpi."

A hundred dots per inch.

Siehe auch

Notationsreferenz: Abschnitt A.12 [Liste der Sonderzeichen], Seite 703.

3.4 Ausgabe kontrollieren

3.4.1 Notationsfragmente extrahieren

Es ist möglich, kleine Abschnitte einer großen Partitur direkt aus der Quelldatei zu erzeugen. Das kann damit verglichen werden, dass man mit der Schere bestimmte Regionen ausschneidet. Es wird erreicht, indem man die Takte, die ausgeschnitten werden sollen (engl. to clip = ausschneiden), extra definiert. Mit folgender Definition beispielsweise

\layout {
clip-regions
= #(list
  (cons
   (make-rhythmic-location 5 1 2)
   (make-rhythmic-location 7 3 4)))
}


Um diese Funktion auch nutzen zu können, muss LilyPond mit dem Parameter -dclip-systems aufgerufen werden. Die Schnipsel werden als EPS ausgegeben und dann zu PDF und PNG konvertiert, wenn diese Formate auch als Parameter angegeben werden.

Zu mehr Information über Ausgabeformate siehe Abschnitt “lilypond aufrufen” in Anwendungsbemutzung.

3.4.2 Korrigierte Musik überspringen

Wenn man Noten eingibt oder kopiert, sind meistens nur die Noten nahe dem Ende (wo gerade neue Noten notiert wurden) wichtig für Kontrolle und Korrektur. Um die Korrektur zu beschleunigen, kann eingestellt werden, dass nur die letzten paar Takte angezeigt werden. Das erreicht man mit dem Befehl

\score { ... }

in der Quelldatei. Damit werden nur die letzten fünf Takte (in einem 4/4-Takt) eines jeden \score-Abschnitts übersetzt. Besonders bei längeren Stücken ist es meistens sehr viel schneller, nur einen kleinen Teil des Stückes zu setzen als die gesamte Länge. Wenn man am Anfang eines Stückes arbeitet (weil etwa ein neuer Teil hinzugefügt werden soll), kann auch die showFirstLength-Eigenschaft nützlich sein.
Nur bestimmte Teile einer Partitur zu überspringen, kann mit der Eigenschaft `Score.skipTypesetting` sehr genau kontrolliert werden. Für den Bereich, für den sie auf „wahr“ gesetzt wird, wird kein Notensatz ausgegeben.

Diese Eigenschaft kann auch benutzt werden, um die MIDI-Ausgabe zu kontrollieren. Hiermit werden alle Ereignisse, auch Tempo- und Instrumentenwechsel ausgelassen. Man muss also sehr genau darauf achten, dass nichts unerwartetes geschieht.

In polyphoner Notation wirkt sich `Score.skipTypesetting` auf alle Stimmen und Systeme aus, sodass noch mehr Zeit bei der Übersetzung der Datei gespart wird.

3.4.3 Alternative Ausgabeformate


3.4.4 Die Notationsschriftart verändern

Gonville ist eine Alternative zu der Emmentaler-Schriftart, die in LilyPond eingesetzt wird und kann unter der Adresse

http://www.chiark.greenend.org.uk/~sgtatham/gonville/ (http://www.chiark.greenend.org.uk/~sgtatham/gonville/)

heruntergeladen werden. Hier einige Takte Noten mit der Gonville-Schriftart:

Und hier einige Beispieltauke in der Feta-Glyphe:

Installationsanweisungen für MacOS

Kapitel 3: Allgemeine Eingabe und Ausgabe

Siehe auch
Handbuch zum Lernen: Abschnitt “Mehr Information” in Handbuch zum Lernen.

Bekannte Probleme und Warnungen

3.5 MIDI-Ausgabe


Der Notensatz von LilyPond kann in MIDI umgewandelt werden, so dass man sich anhören kann, was man notiert hat. Das hilft oft sehr gut bei der Überprüfung: falsche Oktaven oder falsche Versetzungsmarken lassen sich meist sehr gut hören.

Normale MIDI-Ausgabe ist etwas roh; optional kann eine verbesserte und realistischere MIDI-Ausgabe mit einem Abschnitt 3.5.7 [Artikuliere-Skript], Seite 496, erzeugt werden.

Die MIDI-Ausgabe benötigt einen Kanal für jedes System und reserviert Kanal 10 für Schlagzeug. Es gibt nur 16 MIDI-Kanäle pro Gerät, sodass MIDI-Kanäle mehrfach benutzt werden, wenn eine Partitur mehr als 15 Notensysteme hat.

3.5.1 MIDI-Dateien erstellen

Um eine MIDI-Datei aus einer LilyPond-Quelldatei zu erstellen, muss eine \midi-Umgebung zu der \score-Umgebung hinzugefügt werden, etwa so:

```
\score {
  ...
  \midi { }
}
```

Wenn in einer \score-Umgebung nur eine \midi-Umgebung, aber keine \layout-Umgebung vorkommt, wird nur MIDI produziert. Wenn auch die Notation gewünscht ist, muss zusätzlich die \layout-Umgebung vorhanden sein:

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


Das Anfangstempo und spätere Tempoänderungen können mit dem \tempo-Befehl innerhalb der Notation notiert werden. Er bewirkt Tempoänderungen auch in der MIDI-Datei. Der Befehl setzt gleichzeitig auch eine Tempobezeichnung in die Noten, welches aber auch unterdrückt werden kann, siehe [Metronomangabe], Seite 65. Eine andere Möglichkeit, ein eigenes MIDI-Tempo anzugeben, wird weiter unten gezeigt, siehe Abschnitt 3.5.2 [Der MIDI-Block], Seite 489.
Aufgrund einiger Einschränkungen auf Windows ist auf Windows-Systemen die Standarddateierweiterung von MIDI-Dateien .mid. Andere Betriebssysteme verwenden weiterhin .midi. Wenn eine andere Endung erwünscht ist, kann man die folgende Zeile auf oberster Ebene der Quelldatei, vor Beginn eines \book, \bookpart oder \score-Blocks einfügen:

```
%(ly:set-option 'midi-extension "midi")
```

Diese Codezeile setzt die Dateiendung auf .midi.

Als Alternative kann man diese Option auch als Kommandozeilenparameter übergeben:
```
lilypond ... -dmidi-extension=midi lilyDatei.ly
```

**Instrumentenbezeichnungen**

Das MIDI-Instrument, mit dem ein bestimmtes System wiedergegeben werden soll, wird durch die `Staff.midiInstrument`-Eigenschaft bestimmt, die auf eine Instrumentenbezeichnung gesetzt werden muss. Die Bezeichnungen sind aufgelistet in Abschnitt A.6 [MIDI-Instrumente], Seite 622, und müssen in der dort definierten Schreibweise notiert werden.

```
\new Staff {
  \set Staff.midiInstrument = "glockenspiel"
  ...Noten...
}
\new Staff \with {midiInstrument = "cello"} {
  ...Noten...
}
```

Wenn die Schreibweise nicht genau einem definierten Instrument aus der Liste entspricht, wird ein Piano-Klang benutzt ("acoustic grand").

**Ausgewählte Schnipsel**

*Changing MIDI output to one channel per voice*

When outputting MIDI, the default behavior is for each staff to represent one MIDI channel, with all the voices on a staff amalgamated. This minimizes the risk of running out of MIDI channels, since there are only 16 available per track.

However, by moving the `Staff_performer` to the `Voice` context, each voice on a staff can have its own MIDI channel, as is demonstrated by the following example: despite being on the same staff, two MIDI channels are created, each with a different `midiInstrument`.

```
\score {\new Staff <<
  \new Voice \relative c''' {\set midiInstrument =="#flute"
    \voiceOne
    \key g \major
    \time 2/2
    r2 g -"Flute" ~
    g fis ~
    fis4 g8 fis e2 ~
    e4 d8 cis d2
  }
  \new Voice \relative c' {\set midiInstrument =="#clarinet"
    \voiceTwo
    b1 -"Clarinet"
    a2. b8 a
```
Kapitel 3: Allgemeine Eingabe und Ausgabe

Bekannte Probleme und Warnungen

Veränderungen der MIDI-Lautstärke sind nur effektiv, wenn sie zu Beginn einer Note angefordert werden, sodass die Lautstärke während einer Notendauer nicht geändert werden kann.

Nicht alle MIDI-Spieler können Tempoänderungen richtig wiedergeben. Spieler, die hierzu in der Lage sind, sind unter Anderen MS Windows Media Player und tididity (http://timidity.sourceforge.net/).

3.5.2 Der MIDI-Block

Eine \midi-Umgebung muss innerhalb von einer \score-Umgebung vorkommen, wenn MIDI-Ausgabe gewünscht ist. Sie entspricht der layout-Umgebung, aber ist etwas einfacher aufgebaut. Oft wird die MIDI-Umgebung einfach leer gelassen, aber hier können auch Kontexte umgeändert werden, neue Kontexte definiert werden oder neue Werte definiert werden. Das folgende Beispiel etwa definiert das MIDI-Tempo, ohne dass in der Partitur eine Metronombezeichnung gesetzt wird:

\score {
  ...Noten...
  \midi {
    \tempo 4 = 72
  }
}

Hier wird das Tempo auf 72 Viertelnoten pro Minute definiert. \tempo ist eigentlich ein Musikbefehl, der die Eigenschaften während der Interpretation der Musik einstellt: im Kontext von Ausgabedefinitionen wie etwa einem \midi-Kontext werden sie neu interpretiert, als ob es sich um Kontextmodifikatoren handelte.

Kontextdefinitionen des \midi-Kontextes entsprechen der Syntax, wie sie in der layout-Umgebung benutzt wird. Klangübersetzungsmodule werden performer genannt. Die Kontexte
für die MIDI-Ausgabe sind in der Datei ../ly/performer-init.ly definiert, siehe Abschnitt "Mehr Information" in Handbuch zum Lernen. Um beispielsweise die Auswirkung von Dynamikzeichen aus der MIDI-Ausgabe zu entfernen, müssen folgende Zeilen eingefügt werden:

```latex
\midi {
  ...
  \context {
    \Voice
    \remove "Dynamic_performer"
  }
}
```

Eine MIDI-Ausgabe wird nur erstellt, wenn die \midi-Umgebung in eine Partiturumgebung eingefügt wird, die mit dem Befehl \score beginnt.

```latex
\score {
  {...Noten...}
  \midi { }
}
```

3.5.3 Was geht in die MIDI-Ausgabe

In MIDI unterstützt

Die folgenden Notationselemente werden in die MIDI-Ausgabe aufgenommen:

- Tonhöhen
- Mikrotöne (siehe [Versetzungszeichen], Seite 5. Für die Ausgabe wird ein Spieler benötigt, der Tonhöhen verändern kann.)
- Akkorde, die als Symbole notiert werden
- Rhythmen, die als Dauern notiert sind, inklusive N-tolen
- Tremolo, das ohne \;:[Zahl]\; notiert ist
- Überbindungen
- Dynamikzeichen
- Crescendi, decrescendi zu mehreren Noten
- Tempoänderungen, die mit einer Tempo-Bezeichnung eingegeben werden
- Gesangstext

Durch Einsatz vom Abschnitt 3.5.7 [Artikuliere-Skript], Seite 496, können noch einige Elemente zu der Liste hinzugefügt werden:

- Artikulationen (Bögen, Staccato usw.)
- Triller usw.
- Rallentando und accelerando

In MIDI nicht unterstützt

Folgende Notationselemente werden nicht in die MIDI-Ausgabe einbezogen, außer am setzt das Abschnitt 3.5.7 [Artikuliere-Skript], Seite 496, ein:

- Rhythmus, der als Anmerkung notiert wird, bspw. Swing
- Temperänderungen, die als Anmerkung ohne Tempobezeichnung notiert werden
- Staccato und andere Artikulationen und Ornamente
- Legato- und Phrasierungsbögen
- Crescendi, decrescendi zu einer einzelnen Note
Kapitel 3: Allgemeine Eingabe und Ausgabe

3.5.4 Wiederholungen im MIDI

Mit einigen Veränderungen im Notentext können alle Wiederholungstypen auch in der MIDI-Ausgabe wiedergegeben werden. Das wird erreicht, indem die \unfoldRepeats-Funktion eingesetzt wird. Diese Funktion verändert alle Wiederholungen in ausgeschriebene Noten.

\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 "|." 

In Paraturen mit mehreren Stimmen funktioniert das Ausschreiben der Wiederholungen im MIDI nur richtig, wenn jede Stimme vollständig notierte Wiederholungsanweisungen enthält.

Wenn eine Partitur mit diesem \unfoldRepeats-Befehl erstellt wird, ist er notwendig, zwei \score-Umgebungen einzurichten: in der einen werden die Wiederholungen ausgeschrieben und nur eine MIDI-Ausgabe produziert, in der anderen werden die Wiederholungen notiert und als Partitur gesetzt. Das Beispiel gibt einen Hinweis, wie eine derartige Datei aussehen kann:

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

3.5.5 MIDI-Lautstärke kontrollieren


Dynamik-Zeichen

Dynamikanweisungen werden als ein bestimmter Bruch der insgesamt zur Verfügung stehenden MIDI-Lautstärke notiert. Die Standardbrüche reichen von 0,25 für ppppp bis hin zu 0,95 für ffff. Diese Anweisung befinden sich in der Datei ../scm/midi.scm, siehe auch Abschnitt “Mehr Information” in Handbuch zum Lernen. Diese Brüche können nach Belieben geändert oder erweitert
werden, indem eine Funktion erstellt wird, die ein Dynamikzeichen als Argument nimmt und den erforderlichen Bruch ausgibt; schließlich muss noch \texttt{Score.dynamicAbsoluteVolumeFunction} auf diese Funktion gesetzt werden.

Beispielhaft soll gezeigt werden, wie man eine \textit{Rinforzando}-Dynamik, \texttt{rfz}, auch in die MIDI-Ausgabe übernehmen kann. Gleiches gilt für neue, selbstdefinierte Dynamikzeichen, die in den Standarddefinitionen nicht enthalten sind. Die Scheme-Funktion, die hier definiert wird, setzt den Bruch von 0,9 für eine \texttt{rfz}-Anweisung und ruft andernfalls die Standardanweisungen auf:

\begin{verbatim}
    #(define (myDynamics dynamic)
        (if (equal? dynamic "rfz")
            0.9
            (default-dynamic-absolute-volume dynamic))
    
    \score {
        \new Staff {
            \set Staff.midiInstrument = "cello"
            \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
            \new Voice {
                \relative {a'4 \pp b c- \rfz}
            }
        }
        \layout {}
        \midi {}
    }
\end{verbatim}

Alternativ, insbesondere wenn die gesamte Tabelle der MIDI-Lautstärken umdefiniert werden soll, ist es besser, die \textit{default-dynamic-absolute-volume-Prozedur in der Datei} \texttt{../scm/midi.scm} und die hiermit verknüpfte Tabelle als Modell zu benutzen. Das letzte Beispiel dieses Abschnittes zeigt, wie das gemacht werden kann.

**MIDI-Lautstärke**

Die generellen Mindest- und Höchstwerte für die Lautstärke der MIDI-Datei wird kontrolliert, indem die Eigenschaften \texttt{midiMinimumVolume} und \texttt{midiMaximumVolume} auf der Score-Ebene gesetzt werden. Diese Eigenschaften haben nur Einfluss auf Dynamikzeichen, sodass ein Dynamikzeichen direkt an den Anfang der Partitur gestellt werden muss, wenn diese Einstellung von Anfang an Wirkung zeigen soll. Der Bruch, der dann den einzelnen Dynamikzeichen entspricht, wird mit der Formel

\[ \text{midiMinimumVolume} + (\text{midiMaximumVolume} - \text{midiMinimumVolume}) \times \text{Bruch} \]

errechnet. Im folgenden Beispiel wird die generelle MIDI-Lautstärke auf den Bereich zwischen 0,2 und 0,5 eingeschränkt.

\begin{verbatim}
\score {
    <<
        \new Staff {
            \key g \major
            \time 2/2
        }
    }
\end{verbatim}
Verschiedene Instrumente angleichen (i)

Wenn die Mindest- und Höchstwerte für die MIDI-Lautstärke innerhalb eines Staff-Kontextes gesetzt werden, kann damit die relative Lautstärke einzelner Instrumente kontrolliert werden. Damit kann man die Qualität der MIDI-Datei merklich verbessern.

In diesem Beispiel wird die Lautstärke der Klarinette relativ zur Lautstärke der Flöte verringert. In jeder Stimme muss eine Dynamikanweisung für die erste Note gesetzt werden, damit diese Einstellung korrekt funktioniert.
Verschiedene Instrumente angleichen (ii)

Wenn Mindest- und Höchstwerte für die Lautstärke der MIDI-Datei nicht vorgegeben werden, nimmt LilyPond standardmäßig einige Anpassungen für die Lautstärken bestimmter Instrumente vor. Diese Instrumente und ihre entsprechende Veränderung lassen sich aus der Tabelle `instrument-equalizer-alist` in der Datei `../scm/midi.scm` entnehmen.


Das folgende Beispiel definiert für die Flöte und Klarinette relative Lautstärkewerte, die denen des vorigen Beispiels entsprechen.

```lisp
#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist

(append
```
3.5.6 Schlagzeug in MIDI

Einige Instrumente, die keine diskreten Tonhöhen haben, können nicht über den MIDI-Kanal 10 erreicht werden und müssen deshalb in einem normalen Staff-Kontext mit passenden normalen Tonhöhen notiert werden. Es handelt sich um melodic tom, taiko drum, synth drum u. a.


**Bekannte Probleme und Warnungen**

Weil der MIDI-Standard keine Peitschenschläge kennt, wird ein Schlagstock (sidestick) für diesen Zweck eingesetzt.

### 3.5.7 Artikulier-Skript


Um das Artikulier-Skript einzusetzen, muss oben in der Eingabedatei

```latex
\include "articulate.ly"
```

eingefügt werden. Im `\score`-Abschnitt schreibt man

```latex
\unfoldRepeats \articulate <<
\der Rest der Partitur...
>>
```

Wenn die Eingabedatei auf diese Art verändert wurde, wird die Notenausgabe stark verändert, aber die MIDI-Ausgabe produziert ein besseres Ergebnis.

Wenn auch nicht unbedingt notwendig, damit das Artikulier-Skript funktioniert, bietet es sich an, \unfoldRepeats wie im Beispiel oben zu verwenden, weil dadurch Abkürzungen wie etwa Triller ausgeführt werden.

**Bekannte Probleme und Warnungen**

Articulate verkürzt Akkorde und manche Musik (besonders Orgelmusik) kann schlechter klingen.

### 3.6 Musikalische Information extrahieren

Neben graphischer Ausgabe und MIDI kann LilyPond auch die musikalische Information als Text anzeigen:

#### 3.6.1 LilyPond-Notation anzeigen

Mit der musikalischen Funktion `\displayLilyMusic` kann man einen musikalischen Ausdruck anzeigen. Um die Ausgabe zu sehen, wird LilyPond üblicherweise auf der Kommandozeile aufgerufen. Beispielsweise

```latex
{ \displayLilyMusic \transpose c a, { c4 e g a bes } }
```

zeigt an:

```latex
{ a,4 cis e fis g }
```

Standardmäßig gibt LilyPond diese Nachrichten auf die Kommandozeile aus, zusammen mit all den anderen LilyPond-Nachrichten über die Kompilation. Um die Nachrichten zu speichern, kann man die Ausgabe in eine Datei umleiten:

```bash
lilypond file.ly >display.txt
```
LilyPond zeit nicht nur die musikalischen Ausdrücke an, sondern interpretiert sie auch (weil \displayLilyMusic sie an das Programm zurückgibt, zusätzlich zur Anzeige). Das ist sehr praktisch, denn man kann einfach \displayLilyMusic in vorhandene Noten einfügen, um Informationen darüber zu erhalten. Wenn Sie nicht wollen, dass LilyPond die angezeigten musikalischen Ausdrücke auch interpretiert, muss \void eingesetzt werden, damit der Ausdruck für die Interpretation ignoriert wird:

\{
\void \displayLilyMusic \transpose c a, { c4 e g a bes }
\}

### 3.6.2 Musikalische Scheme-Ausdrücke anzeigen

Siehe Abschnitt “Musikalische Funktionen darstellen” in Extending.

### 3.6.3 Musikalische Ereignisse in einer Datei speichern

Musikereignisse können in einer Datei Notensystem für Notensystem gespeichert werden, indem eine Datei in die Paritur eingefügt wird:

\include "event-listener.ly"

Das erstellt Dateien mit der Bezeichnung DATEINAME-SYSTEMNAME.notes oder DATEINAME-unnamed-staff.notes für jedes Notensystem. Wenn mehrere Systeme ohne Bezeichnung auftreten, werden die Ereignisse aller Notensysteme zusammen in der selben Datei gemischt. Die Ausgabe sieht folgendermaßen aus:

```
0.000 note 57 4 p-c 2 12
0.250 note 62 4 p-c 7 12
0.500 note 66 8 p-c 9 12
0.625 note 69 8 p-c 14 12
0.750 rest 4
0.750 breathe
```

Die Syntax ist eine durch Tabulatoren getrennte Zeile mit zwei festen Zellen, gefolgt von optionalen Parametern.

```
time type ...params...
```

Diese Information kann einfach ein ein anderes Programm wie etwa ein Python-Skript eingefügt werden und kann nützlich für Forscher sein, die musikalische Analyse- oder Wiedergabeexperimente mit LilyPond machen wollen.

### Bekannte Probleme und Warnungen

Nicht alle musikalischen Ereignisse werden von event-listener.ly unterstützt. Es handelt sich eher um ein gut gemachtes „proof of concept“. Wenn Ereignisse, die Sie brauchen, nicht in enthalten sind, können Sie event-listener.ly in Ihr LilyPond-Verzeichnis kopieren und die Datei verändern, sodass sie die benötigte Information ausgibt.
4 Abstände


Die Verteilung der Musik auf der Seite geschieht grob gesagt in vier Schritten. Zuerst werden flexible Entfernungen („springs“) gewählt, die auf den Notendauern basieren. Alle möglichen Zeilenumbrüche werden getestet und ein „Schlechtigkeitsscore“ für die Umbrüche erstellt. Danach wird die mögliche Höhe eines Systems ermittelt und schließlich wird eine bestimmte Kombination aus Seiten- und Zeilenumbruch ausgewählt, sodass weder die horizontale noch die vertikale Platzverteilung zu eng oder zu weit gesetzt wird.


4.1 Seitenlayout

Dieser Abschnitt behandelt Seitenlayout-Optionen innerhalb der \paper-Umgebung.

4.1.1 Die \paper-Umgebung

Die \paper-Umgebung kann innerhalb einer \book-, nicht aber innerhalb einer \score-Umgebung vorkommen. Einstellungen in \paper wirken sich auf das gesamte Buch aus, welches viele einzelne Partituren beinhalten kann. Einstellungen, die in der \paper-Umgebung vorkommen können, beinhalten:

• die set-paper-size-Scheme-Funktion,
• \paper-Variablen, die zum Verändern des Seitenlayouts eingesetzt werden und
• Beschriftungsdefinitionen, mit denen das Layout von Kopf- und Fußleisten sowie Titeln beeinflusst wird.


Die meisten \paper-Variablen funktionieren nur innerhalb der \paper-Umgebung. Die wenigen, die auch in der \layout-Umgebung funktionieren, finden sich in Abschnitt 4.2.1 [Die layout-Umgebung], Seite 509.

Außer wenn anders angegeben, werden alle \paper-Variablen, die Abständen auf der Seite entsprechen, in Millimetern gemessen, es sei denn, eine andere Maßeinheit ist definiert. Beispielsweise wird mit folgender Definition der obere Rand (top-margin) 10 mm breit definiert:

\begin{verbatim}
\paper {
  top-margin = 10
}
\end{verbatim}

Damit etwa 0.5 Zoll benutzt werden, muss \in dem Maß nachgestellt werden:

\begin{verbatim}
\paper {
\end{verbatim}
Mögliche Maßeinheiten sind \texttt{mm}, \texttt{cm}, \texttt{in} und \texttt{pt}. Diese Maßeinheiten sind einfach Werte, um von Millimetern zu konvertieren, sie sind in \texttt{ly/paper-defaults-init.ly} definiert. Um Missverständnisse zu vermeiden, wird normalerweise \texttt{mm} geschrieben, auch wenn es eigentlich nicht notwendig wäre.

Man kann die \texttt{paper}-Werte auch mit Scheme definieren. Die Scheme-Entsprechung der obigen Definition ist:

\begin{verbatim}
\paper {
  #(define top-margin (* 0.5 in))
}
\end{verbatim}

Siehe auch

Notationsreferenz Abschnitt 4.1.2 [Papierformat und automatische Skalierung], Seite 499, Abschnitt 3.2.2 [Eigene Kopf und Fußzeilen sowie Titel], Seite 460, Abschnitt 4.2.1 [Die \texttt{layout}-Umgebung], Seite 509.

Installierte Dateien: \texttt{ly/paper-defaults-init.ly}.

4.1.2 Papierformat und automatische Skalierung

Das Papierformat einstellen

\texttt{A4} ist der Standardwert, wenn keine ausdrückliche Angabe gemacht wird. Es gibt jedoch zwei Funktionen, mit denen man das Papierformat ändern kann: \texttt{set-default-paper-size} und \texttt{set-paper-size}.

\begin{verbatim}
(set-default-paper-size "quarto")
\end{verbatim}

welcher immer auf oberster Ebene der Datei geschrieben werden muss, und

\begin{verbatim}
(set-paper-size "tabliod")
\end{verbatim}

welcher in eine \texttt{paper}-Umgebung geschrieben werden muss.

Wenn die \texttt{set-default-paper-size}-Funktion auf oberster Ebene der Datei eingesetzt wird, muss sie vor allen \texttt{paper}-Umgebungen kommen. \texttt{set-default-paper-size} definiert das Papierformat für alle Seiten, während \texttt{set-paper-size} nur das Format der Seiten bestimmt, auf die sich die \texttt{paper}-Umgebung bezieht. Wenn beispielsweise die \texttt{paper}-Umgebung oben in der Datei steht, dann bezieht sie sich auf alle Seiten in der Datei. Wenn die \texttt{paper}-Umgebung innerhalb einer \texttt{book}-Umgebung steht, dann bezieht sie sich nur auf das eine Buch.

Wenn die \texttt{set-paper-size}-Funktion eingesetzt wird, muss sie vor allen anderen Funktionen stehen, die in derselben \texttt{paper}-Umgebung benutzt werden. Siehe [Automatische Skalierung auf ein Papierformat], Seite 500.


Der folgende Befehl kann benutzt werden, um ein eigenes Papierformat hinzuzufügen, welches dann mit \texttt{set-default-paper-size} oder \texttt{set-paper-size} benutzt werden kann:

\begin{verbatim}
(set! paper-alist (cons '("mein Format" . (cons (* 15 in) (* 3 in))) paper-alist))
\end{verbatim}

\begin{verbatim}
\paper {
  #(set-paper-size "mein Format")
}
\end{verbatim}

Die Einheiten in (Fuß), cm (Centimeter) und mm (Millimeter) können benutzt werden.
Wenn das Symbol 'landscape an die Funktion \texttt{set-default-paper-size} gehängt wird, werden die Seiten um 90° gedreht und die Notensysteme entsprechend breiter gesetzt.

\begin{verbatim}
#(set-default-paper-size "a6" 'landscape)
\end{verbatim}

Die Notenausgabe wird \textit{nicht} gedreht, nur das Papierformat.

\section*{Siehe auch}

Notationsreferenz: [Automatische Skalierung auf ein Papierformat], Seite 500, Abschnitt A.5 [Vordefinierte Papierformate], Seite 619.

Installierte Dateien: \texttt{scm/paper.scm}.

\section*{Automatische Skalierung auf ein Papierformat}

Wenn das Papierformat mit einer der Scheme-Funktionen (\texttt{set-default-paper-size} oder \texttt{set-paper-size}) geändert wird, werden die Werte einiger \texttt{paper}-Variablen automatisch an die neue Größe angepasst. Um die automatische Skalierung für eine bestimmte Variable zu umgehen, kann die Variable definiert werden, nachdem man das Papierformat angegeben hat. Es sollte beachtet werden, dass die automatische Anpassung nicht ausgelöst wird, wenn man nur die \texttt{paper-height} oder \texttt{paper-width}-Variablen verändert, obwohl \texttt{paper-width} andere Werte beeinflussen kann (das muss von der automatischen Skalierung unterschieden werden und wird unten behandelt). Die Funktionen \texttt{set-default-paper-size} und \texttt{set-paper-size} werden behandelt in [Das Papierformat einstellen], Seite 499.

Die vertikalen Dimensionen, die durch die automatische Skalierung verändert werden sind: \texttt{top-margin} und \texttt{bottom-margin} (siehe Abschnitt 4.1.3 [Vertikale \texttt{paper}-Variablen mit festen Abständen], Seite 500). Die horizontalen Dimensionen, die durch die automatische Skalierung verändert werden, sind \texttt{left-margin}, \texttt{right-margin}, \texttt{inner-margin}, \texttt{outer-margin}, \texttt{binding-offset}, \texttt{indent} und \texttt{short-indent} (siehe Abschnitt 4.1.5 [\texttt{paper}-Variablen für horizontale Abstände], Seite 503).

Die Standardwerte für diese Dimensionen sind in \texttt{ly/paper-defaults-init.ly} definiert, wobei interne Variablen mit den Bezeichnungen \texttt{top-margin-default}, \texttt{bottom-margin-default}, usw. benutzt werden. Das sind die Werte für die Standardpapiergröße a4. Zum Vergleich: a4 hat Werte von 297\texttt{\textbackslash mm} für \texttt{paper-height} und 210\texttt{\textbackslash mm} für \texttt{paper-width}.

\section*{Siehe auch}

Notationsreferenz: Abschnitt 4.1.3 [Vertikale \texttt{paper}-Variablen mit festen Abständen], Seite 500, Abschnitt 4.1.5 [\texttt{paper}-Variablen für horizontale Abstände], Seite 503.

Installierte Dateien: \texttt{ly/paper-defaults-init.ly}, \texttt{scm/paper.scm}.

\subsection*{4.1.3 Vertikale \texttt{paper}-Variablen mit festen Abständen}

\begin{itemize}
\item \textbf{Achtung:} Einige \texttt{paper}-Dimensionen werden automatisch nach Papierformat skaliert, was zu ungewolltem Verhalten führen kann. Siehe [Automatische Skalierung auf ein Papierformat], Seite 500.
\end{itemize}

Standardwerte (vor der Skalierung) sind definiert in \texttt{ly/paper-defaults-init.ly}.

\item \texttt{paper-height} Die Höhe der Seite, standardmäßig nicht definiert. Die automatische Skalierung einiger vertikaler Dimensionen wird hiervon nicht betroffen.
Kapitel 4: Abstände

**top-margin**
Der Rand zwischen dem oberen Ende der Seite und dem oberen Ende des bedruckbaren Bereichs. Wenn das Papierformat verändert wurde, wird der Standardwert dieser Dimension entsprechend skaliert.

**bottom-margin**
Der Rand zwischen dem unteren Ende der Seite und dem unteren Ende des bedruckbaren Bereichs. Wenn das Papierformat verändert wurde, wird der Standardwert dieser Dimension entsprechend skaliert.

**ragged-bottom**
Wenn auf wahr gesetzt, werden die Systeme nicht vertikal bis zum unteren Seitenrand verteilt. Sollte auf wahr gesetzt sein für Stücke, die nur ein bis zwei Notensystemgruppen pro Seite haben, etwa Orchesterpartituren.

**ragged-last-bottom**
Wenn auf falsch gesetzt, werden die Systeme vertikal auf der letzten Seite verteilt. Bei Stücken, die grob zwei oder mehr Seiten füllen, sollten es auf falsch (*false*) gesetzt werden. Hiermit wird auch die letzte Seite von Teilen eines `\book`, die mit `\bookpart` erstellt sind, beeinflusst.

Siehe auch

Notationsreferenz: [Automatische Skalierung auf ein Papierformat], Seite 500.
Installierte Dateien: `ly/paper-defaults-init.ly`
Schnipsel: Abschnitt “Spacing” in Schnipsel.

Bekannte Probleme und Warnungen

Die Titel (aus der `\header`-Umgebung) werden als Systemgruppe behandelt, sodass `ragged-bottom` und `ragged-last-bottom` auch zusätzlichen Abstand zwischen den Titel und dem ersten System einer Paritur einfügt.

Explicite definierte Papierformate überschreiben alle vom Benutzer erstellte Randeneinstellung für die Ränder oben und unten.

4.1.4 Vertikale `\paper`-Variablen mit flexiblen Abständen

In den meisten Fällen bietet es sich an, dass die vertikalen Abstände zwischen bestimmten Objekten (wie Ränder, Titel, Notensystemgruppen und einzelne Partituren) flexibel gehalten werden, sodass sie je nach Situation gedehnt oder komprimiert werden können. Es gibt eine Anzahl von Variablen für die `\paper`-Umgebung, mit denen man das Dehnungsverhalten dieser Dimensionen beeinflussen kann. Sie finden sich unten aufgelistet.

Dabei sollte beachtet werden, dass die Variablen, die in diesem Abschnitt behandelt werden, nicht die Platzierung und das Dehnungsverhalten von Notensystemen innerhalb der einzelnen Systemgruppen behandeln. Die Dehnung zwischen Notensystemen wird mit Grob-Eigenschaften kontrolliert, deren Einstellungen normalerweise innerhalb der `\score`-Umgebung vorgenommen werden, und nicht innerhalb der `\paper`-Umgebung. Siehe auch Abschnitt 4.4.1 [Flexible vertikale Abstände in Systemgruppen], Seite 520.

Struktur der Alisten für flexible vertikale Abstände

Jede der flexiblen vertikalen Abstandsvariablen ist eine Aliste (eine assoziative Liste), die vier Schlüssel (engl. key) enthält:

- **basic-distance** (Grund-Abstand) – der vertikale Abstand, gemessen in Systemzwischenräumen, zwischen den **Referenzpunkten** zweier Elemente, wenn keine Zusammenstöße vorkommen würden und keine Dehnung oder Kompression stattfindet. Der Referenzpunkt
Kapitel 4: Abstände

502
einer (Titel-)Beschriftung (auf höchster Ebene) ist sein höchster Punkt und der Referenzpunkt einer Systemgruppe ist der vertikale Mittelpunkt des nächsten StaffSymbol – sogar wenn eine Nicht-Notensystemzeile (wie etwa ein Lyrics-Kontext) dazwischen kommt. Werte für basic-distance, die weniger als entweder padding oder minimum-distance sind, haben keine Bedeutung, weil der sich daraus ergebende Abstand niemals weniger als entweder padding oder minimum-distance ergibt.

• minimum-distance (minimaler Abstand) – der kleinste erlaubte vertikale Abstand, gemessen in Systemzwischenräumen, zwischen den Referenzpunkten der zwei Elemente, wenn Kompression stattfindet. Werte für minimum-distance, die geringer als padding sind, haben keine Bedeutung, weil der sich daraus ergebende Abstand niemals weniger als padding ergibt.

• padding (Verschiebung) – der minimal benötigte vertikale blanke Freiraum zwischen den Bounding-Boxen (oder Skyline) der zwei Objekten, gemessen in Notenlinienabständen.

• stretchability (Dehnbarkeit) – ein einheitsloses Maß der Leichtigkeit, mit der sich die Dimension dehnen lässt (ohne dass Zusammenstöße auftreten). Wenn es null ist, wird der Abstand nicht gedehnt (außer ein Zusammenstoß würde auftreten), wenn es positiv ist, hängt die Wichtigkeit der Dehnbarkeit eines bestimmten Objekts nur noch von seiner Beziehung zu dem Wert des anderen Objekts ab. Beispielsweise wemn eine Dimension die doppelte Dehnbarkeit als die andere hat, wird sie auch zweimal so einfach gedehnt. Werte sollten nicht-negativ und reale Zahlen sein. Der Wert +inf.0 ruft einen programming_error hervor und wird ignoriert, aber 1.0e7 kann für einen so gut wie unendlich dehnbaren Abstand eingesetzt werden. Wenn der Wert nicht gesetzt wird, ist der Standardwert der von basic-distance. Die Wahrscheinlichkeit einer Dimension, sich zu verkleinern, kann man nicht direkt beeinflussen, sondern sie ergibt sich aus (space − minimum-distance).

Wenn eine Seite einen nicht ausgeglichenen unteren Rand hat, ist der resultierende Abstand der größte von:

• basic-distance,
• minimum-distance und
• padding plus der kleinste nötige Abstand, um Zusammenstöße zu vermeiden.

Bei Partituren über mehrere Seiten mit nicht ausgeglichem unteren Rand greift die letzte Seite auf die gleiche Positionierung zurück wie die vorhergehende Seite, vorausgesetzt, dafür ist genügend Platz vorhanden.

Spezifische Methoden, um Alisten zu verändern, werden behandelt in Abschnitt 5.3.6 [Alisten verändern], Seite 578. Das folgende Beispiel demonstriert beide Arten, wie diese Alisten verändert werden können. Der erste Aufruf verändert nur einen Schlüsselwert einzeln, während der zweite die Variable vollständig neu definiert:

```
paper {
  system-system-spacing.basic-distance = #8
  score-system-spacing =
    #'(basic-distance . 12)
    (minimum-distance . 6)
    (padding . 1)
    (stretchability . 12))
}
```

Liste der flexiblen vertikalen Abstandsvariablen in \paper

Die Bezeichnungen dieser Variablen entsprechen dem Format obere-untere-platzierung, wobei obere und untere die zu platzierenden Elemente darstellen. Jeder Abstand wird zwischen den Referenzpunkten der beiden Elemente gemessen (siehe Beschreibung der Alistenstruktur
In diesen Variablenbezeichnungen bedeutet ‘markup’ (Beschriftung) sowohl Titelbeschriftungen (bookTitleMarkup oder scoreTitleMarkup) als auch Beschriftungen auf höchster Ebene (siehe Abschnitt 3.1.5 [Die Dateistruktur], Seite 451). Alle Entfernungen werden in Systemzwischenräumen gemessen.


- **markup-system-spacing**: der Abstand zwischen einer (Titel-)Beschriftung (auf höchster Ebene) und der darauf folgenden Systemgruppe.
- **score-markup-spacing**: der Abstand zwischen dem letzten System einer Partitur und der darauf folgenden (Titel-)Beschriftung (auf höchster Ebene).
- **score-system-spacing**: der Abstand zwischen dem letzten System einer Partitur und dem ersten System der folgenden Partitur, wenn keine (Titel-)Beschriftung (auf höchster Ebene) dazwischen vorkommt.
- **system-system-spacing**: der Abstand zwischen zwei Systemgruppen der selben Partitur.
- **markup-markup-spacing**: der Abstand zwischen zwei (Titel-)Beschriftungen (auf höchster Ebene).
- **last-bottom-spacing**: der Abstand vom letzten System oder Beschriftung auf höchster Ebene auf einer Seite zum unteren Rand des bedruckbaren Bereichs (also bis zum Anfang des unteren Randes).
- **top-system-spacing**: der Abstand zwischen dem oberen Rand des bedruckbaren Bereichs (also dem Ende des oberen Randes) und dem ersten System auf der Seite, wenn keine (Titel-)Beschriftung (auf höchster Ebene) dazwischen kommt.
- **top-markup-spacing**: der Abstand vom oberen Rand des bedruckbaren Bereichs (also dem Ende des oberen Randes) zur ersten (Titel-)Beschriftung (auf höchster Ebene) auf einer Seite, wenn keine Systemgruppe dazwischen kommt.

Siehe auch

Notationsreferenz: Abschnitt 4.4.1 [Flexible vertikale Abstände in Systemgruppen], Seite 520.
Schnipsel: Abschnitt “Spacing” in Schnipsel.

### 4.1.5 \paper-Variablen für horizontale Abstände


\textbf{\textbackslash paper-Variablen für Breite und Ränder}

Standardwerte (vor der Skalierung), die hier nicht aufgelistet sind, finden sich in \texttt{ly/paper-defaults-init.ly}.

\textbf{paper-width} \hfill

Die Breite der Seite, standardmäßig nicht definiert. Während \texttt{paper-width} keine Auswirkungen auf die automatische Skalierung einiger horizontaler Dimensionen hat, beeinflusst es dennoch die \texttt{line-width}-Variable. Wenn sowohl \texttt{paper-width} als auch \texttt{line-width} definiert sind, dann werden auch \texttt{left-margin} und \texttt{right-margin} aktualisiert. Siehe auch \texttt{check-consistency}.

\textbf{line-width} \hfill

Die horizontale Ausdehnung der Notenlinien in nicht-eingerückten, Systemen mit Ausgleich zum rechten Rand, entspricht \((\texttt{paper-width} - \texttt{left-margin} - \texttt{right-margin})\) wenn nicht definiert. Wenn \texttt{line-width} definiert ist und sowohl \texttt{left-margin} als auch \texttt{right-margin} nicht definiert sind, dann werden die Ränder aktualisiert, sodass die Systeme mittig auf der Seite zentriert werden. Siehe auch \texttt{check-consistency}. Diese Variable kann auch in der \texttt{\layout}-Umgebung definiert werden.

\textbf{left-margin} \hfill

Der Rand zwischen der linken Papierkante und dem Beginn der Systeme ohne Einrückungen. Wenn das Papierformat verändert wird, wird auch der Standardwert dieser Dimension entsprechend skaliert. Wenn \texttt{left-margin} nicht definiert ist und sowohl \texttt{line-width} als auch \texttt{right-margin} definiert sind, dann wird \texttt{left-margin} auf den Wert \((\texttt{paper-width} - \texttt{line-width} - \texttt{right-margin})\) gesetzt. Wenn nur \texttt{line-width} definiert ist, dann werden beide Ränder auf den Wert \((\texttt{paper-width} - \texttt{line-width}) / 2\) gesetzt und die Systeme demzufolge auf der Seite zentriert. Siehe auch \texttt{check-consistency}.

\textbf{right-margin} \hfill

Der Rand zwischen der rechten Papierkante und dem Ende der Systeme mit Randausgleich („Blöckchen“). Wenn das Papierformat geändert wird, wird auch der Standardwert dieser Dimension entsprechend skaliert. Wenn \texttt{right-margin} nicht definiert ist und sowohl \texttt{line-width} als auch \texttt{left-margin} definiert sind, dann wird \texttt{right-margin} auf den Wert \((\texttt{paper-width} - \texttt{line-width} - \texttt{left-margin})\) gesetzt. Wenn nur \texttt{line-width} definiert ist, dann werden beide Ränder auf den Wert \((\texttt{paper-width} - \texttt{line-width}) / 2\) gesetzt und die Systeme demzufolge auf der Seite zentriert. Siehe auch \texttt{check-consistency}.

\textbf{check-consistency} \hfill

Wenn wahr, wird eine Warnung ausgegeben, sollten \texttt{left-margin}, \texttt{line-width} und \texttt{right-margin} zusammen nicht exakt den Wert von \texttt{paper-width} ergeben, und die Werte (außer \texttt{paper-width}) mit ihren Standardwerten belegt (wenn nötig auf das entsprechende Papierformat skaliert). Wenn falsch werden derartige Inkonsistenzen ignoriert und die Systeme dürfen auch über den Seitenrand hinausragen.

\textbf{ragged-right} \hfill

Wenn wahr, werden Notensysteme nicht über die gesamte Zeilenbreite gestreckt, sondern sie enden horizontal entsprechend den enthaltenen Noten. Standard: \texttt{#t} (wahr) für Partituren mit einem System und \texttt{#f} (falsch) für Partituren mit zwei oder mehr Systemen. Diese Variable kann auch in der \texttt{\layout}-Umgebung definiert werden.
Kapitel 4: Abstände 505

ragged-last
Wenn wahr, wird das letzte Notensystem einer Partitur nicht über die gesamte Zeilenbreite gestreckt, sondern es endet horizontal entsprechend den enthaltenen Noten. Standard: #f (falsch). Diese Variable kann auch in der \layout-Umgebung definiert werden.

Siehe auch
Notationsreferenz: [Automatische Skalierung auf ein Papierformat], Seite 500.

Bekannte Probleme und Warnungen
Explozit definierte Papierformate überschreiben alle vom Benutzer erstellte Randeinstellung für die Ränder oben und unten.

\paper-Variablen für zweiseitigen Satz
Standardwerte (vor der Skalierung) sind definiert in ly/paper-defaults-init.ly.

two-sided
Wenn auf wahr (##t) gesetzt, werden inner-margin, outer-margin und binding-offset zusammen benutzt, um die Ränder der Seite in Abhängigkeit von einer geraden oder ungeraden Seitennummer zu errechnen. Damit werden die Werte von left-margin und right-margin überschrieben. Standard: ##f.

inner-margin

outer-margin
Der Rand, den alle Seiten auf der Außenseite haben, wenn sie Teil eines Buches sind. Wenn das Papierformat verändert wird, wird der Standardwert dieser Dimension entsprechend skaliert. Funktioniert nur, wenn two-sided wahr ist.

binding-offset
Der Wert, um welchen inner-margin erhöht wird, um sicherzugehen, dass nichts in der Bindung verschwindet. Wenn das Papierformat verändert wird, wird der Standardwert dieser Dimension entsprechend skaliert. Funktioniert nur, wenn two-sided wahr ist.

Siehe auch
Notationsreferenz: [Automatische Skalierung auf ein Papierformat], Seite 500.

\paper-Variablen für Verschiebungen und Einrückungen
Standardwerte (vor der Skalierung), die hier nicht aufgeführt sind, sind definiert in ly/paper-defaults-init.ly.

horizontal-shift
Der Wert, um den alle Systeme (und auch Überschriften und Systemtrenner) nach rechts verschoben werden. Standard: 0.0\mm.

indent
Der Einzug für das erste System einer Partitur. Wenn das Papierformat verändert wird, wird der Standardwert dieser Dimension entsprechend skaliert. Diese Variable kann auch in der \layout-Umgebung definiert werden.
short-indent
Der Einzug für alle Systeme einer Partitur ausschließlich das erste System. Wenn das Papierformat verändert wird, wird der Standardwert dieser Dimension entsprechend skaliert. Diese Variable kann auch in der \layout-Umgebung definiert werden.

Siehe auch
Notationsreferenz: [Automatische Skalierung auf ein Papierformat], Seite 500.
Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.1.6 Andere \paper-Variablen

\paper-Variablen für den Zeilenumbruch

max-systems-per-page

min-systems-per-page

systems-per-page

system-count

Siehe auch
Notationsreferenz: Abschnitt 4.3.1 [Zeilenumbrüche], Seite 512.

\paper-Variablen für den Seitenumbruch

Standardwerte, die hier nicht aufgelistet sind, finden sich in ly/paper-defaults-init.ly

blank-after-score-page-penalty
Die Strafpunkte, die erteilt werden, wenn eine leere Seite nach einer Partitur und vor der nächsten vorkommt. Der Standardwert hiervon ist kleiner als blank-page-penalty, sodass leere Seiten nach einer Partitur leeren Seiten innerhalb einer Partitur vorgezogen werden.

blank-last-page-penalty
Die Strafpunkte, wenn eine Partitur auf einer ungeraden Seite beendet wird. Standard: 0.

blank-page-penalty


Siehe auch
Notationsreferenz: Abschnitt 4.3.2 [Seitenumbrüche], Seite 514, Abschnitt 4.3.3 [Optimale Seitenumbrüche], Seite 515, Abschnitt 4.3.4 [Optimale Umbrüche zum Blättern], Seite 515, Abschnitt 4.3.5 [Minimale Seitenumbrüche], Seite 516, Abschnitt 4.3.6 [Eine-Seite-Seitenumbrüche], Seite 517.


\paper-Variablen für Seitenzahlen
Standardwerte, die hier nicht aufgelistet sind, finden sich in ly/paper-defaults-init.ly

Der Seitenumbruchsalgorithmus wird davon beeinflusst, ob die erste Seitenzahl gerade oder ungerade ist. Wenn die Variable auf wahr gesetzt wird, entscheidet der Seitenumbruchsalgorithmus selber, ob die Noten auf einer geraden oder ungeraden Seite beginnen sollen. Das hat dann zur Folge, dass die erste Seite entweder bleibt wie sie ist oder um eins erhöht wird. Standard: #f.

Der Wert der Seitenzahl auf der ersten Seite.

Wenn wahr, wird auch auf der ersten Seite die Seitenzahl ausgegeben. Standard: #f.

Wenn falsch, werden Seitenzahlen nicht ausgegeben.

Siehe auch

Ungrade Seitenzahlen befinden sich immer auf der rechten Seite. Wenn Sie die Noten auf Seite 1 beginnen lassen wollen, müssen Sie eine leere Seite nach dem Deckblatt einfügen, damit die Noten auf der rechten Seite mit Seite 1 beginnen.
Verschiedene \paper-Variablen

page-spacing-weight

print-all-headers
Wenn wahr, werden alle Einträge des Titelfeldes (\header-Umgebung) für jede Partitur (\score) ausgegeben. Normalerweise wird nur die Satzbezeichnung und die Opuszahl (piece und opus) ausgegeben. Standard: #f.

system-separator-markup
Ein Beschriftungsobjekt, das zwischen zwei Systeme gesetzt wird. Das wird oft in Orchesterpartituren eingesetzt. Standard: nicht gesetzt. Der Beschriftungsbefehl \slashSeparator, definiert in ly/titling-init.ly, kann für einen Trenner benutzt werden, etwa so:

\(\#(\set-default-paper-size \"a8\")\)

\begin{lyxcode}
\book {
  \paper {
    \system-separator-markup = \\slashSeparator
  }
  \header {
    \tagline = ##f
  }
  \score {
    \relative { c'1 \break c1 \break c1 }
  }
}
\end{lyxcode}

Siehe auch


Schnipsel: Abschnitt “Spacing” in Schnipsel.

Bekannte Probleme und Warnungen
Die Standard-Kopfzeiledefinition setzt die Seitenzahl und das instrument-Feld aus der \header-Umgebung in eine Zeile.
4.2 Partiturlayout

4.2.1 Die \layout-Umgebung

Während die \paper-Umgebung Einstellungen für die Formatierung der Seiten eines gesamten Dokuments enthält, enthält die \layout-Umgebung Einstellungen für einzelne Partituren. Um Layoutoptionen für Partituren global einzustellen, müssen sie in einer \layout-Umgebung gesetzt werden, die sich auf höchster Ebene in der Datei befindet. Um sie für einzelne Partituren festzulegen, muss die \layout-Umgebung innerhalb der \score-Umgebung nach den Noten eingetragen werden. Einstellungen, die in einer \layout-Umgebung vorkommen können, beinhalten:

- die \layout-set-staff-size-Scheme-Funktion,
- Kontextveränderungen in \context-Umgebungen und
- \paper-Variablen, die das Aussehen einer Partitur beeinflussen.

Die \layout-set-staff-size-Funktion wird im nächsten Abschnitt behandelt, Abschnitt 4.2.2 [Die Notensystemgröße einstellen], Seite 511. Kontextveränderungen werden in einem eigenen Kapitel behandelt, siehe Abschnitt 5.1.4 [Umgebungs-Plugins verändern], Seite 558, and Abschnitt 5.1.5 [Die Standardeinstellungen von Kontexten ändern], Seite 560. Die \paper-Variablen, die innerhalb der \layout-Umgebungen erlaubt sind, sind:

- line-width, ragged-right und ragged-last (siehe \paper-Variablen für Breite und Ränder], Seite 504)
- indent und short-indent (siehe \paper-Variablen für Verschiebungen und Einrückungen], Seite 505)
- system-count (siehe \paper-Variablen für den Zeilenumbruch], Seite 506)

Hier ist ein Beispiel für eine \layout-Umgebung:

\layout {
  indent = 2\cm
  \context {
    \StaffGroup
    \override StaffGrouper.staff-staff-spacing.basic-distance = #8
  }
  \context {
    \Voice
    \override TextScript.padding = #1
    \override Glissando.thickness = #3
  }
}

Mehrfache \layout-Umbgebungen können als Ausdrücke auf höchster Ebene eingegeben werden. Das kann beispielsweise nützlich sein, wenn unterschiedliche Einstellungen in extra Dateien gespeichert werden und optional eingefügt werden. Intern wird eine Kopie der aktuellen \layout-Konfiguration gemacht, wenn eine \layout-Umgebung ausgewertet wird, dann erst werden Änderungen aus der Umgebung angewendet und das Ergebnis als die neue aktuelle Konfiguration gespeichert. Aus der Sicht des Benutzers werden die \layout-Umgebungen kombiniert, aber in Konflikt situationen (wenn die gleiche Eigenschaft in unterschiedlichen Umgebungen geändert wird), erhält die spätere Definition den Vorrang.

Wenn also diese Umgebung:

\layout {
  \context {
    \Voice
  }

Kapitel 4: Abstände

\override TextScript.color = #magenta
\override Glissando.thickness = #1.5
}
}
nach der Umgebung aus dem vorherigen Beispiel geschrieben wird, werden die 'padding-
und 'color-Einstellungen für TextScript kombiniert, aber die spätere 'thickness-Einstellung
für Glissando ersetzt (oder versteckt) die vorherige.

\layout-Umgebungen können einer Variable zur späteren Benutzung zugewiesen werden,
aber die Funktionsweise dieser Zuweisung ist geringfügig aber deutlich unterschiedlich von der
Variante, sie auszuschreiben.

Wenn eine Variable etwa so definiert wird:

layoutVariable = \layout {
  \context {
    \Voice
    \override NoteHead.font-size = #4
  }
}

dann enthält sie die aktuelle \layout-Konfiguration mit zusätzlicher Veränderung von NoteHead
'font-size (der Schriftgröße der Notenköpfe), aber diese Kombination wird nicht als nächste
aktuelle Konfiguration gespeichert. Man muss sich im klaren sein, dass die „aktuelle Konfigura-
tion“ gelesen wird, wenn die Variable erstellt wird, nicht wenn sie benutzt wird. Darum ist der
Inhalt der Variable abhängig von ihrer Position in der Partitur.

Die Variable kann dann auch innerhalb einer anderen \layout-Umgebung eingesetzt werden,
etwa:

\layout {
  \layoutVariable
  \context {
    \Voice
    \override NoteHead.color = #red
  }
}

Eine \layout-Umgebung, die eine Variable enthält wie im Beispiel oben, kopiert die aktuelle
Konfiguration nicht, sondern benutzt den Inhalt von \layoutVariable als Basiskonfiguration
für die weiteren Veränderungen. Das heißt, dass Änderungen, die zwischen der Definition der
Variable und ihrer Benutzung definiert wurden, verloren gehen.

Wenn layoutVariable kurz vor ihrer Benutzung definiert wird (oder mit \include eingefügt
wird), entspricht ihr Inhalt der aktuellen Konfiguration plus die Änderungen, die die Variable
definiert. Das obige Beispiel, das den Einsatz von \layoutVariable zeigt, würde in seiner finalen
Version folgende \layout-Umgebung haben:

TextScript.padding = #1
TextScript.color = #magenta
Glissando.thickness = #1.5
NoteHead.font-size = #4
NoteHead.color = #red

plus die Veränderungen an Einrückung (indent) und StaffGrouper.

Aber wenn die Variable vor der ersten \layout-Umgebung definiert wird, würde die aktuelle
Konfiguration nur enthalten:

NoteHead.font-size = #4 % (written in the variable definition)
NoteHead.color = #red % (added after the use of the variable)
Wenn man sorgfältig plant, können \layout-Variablen ein wertvolles Instrument sein, um das Layout-Design von Quellen zu strukturieren und auch dazu dienen, die Layout-Einstellungen an einer bestimmten Stelle wieder zurückzusetzen.

Siehe auch

Notationsreferenz: Abschnitt 5.1.5 [Die Standardeinstellungen von Kontexten ändern], Seite 560, Abschnitt 4.2.2 [Die Notensystemgröße einstellen], Seite 511, Abschnitt 5.1.4 [Umgebungs-Plugins verändern], Seite 558.

Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.2.2 Die Notensystemgröße einstellen

Die Standardgröße der Notensysteme beträgt 20 Punkte (pt). Das kann auf zwei Arten geändert werden:

Um die Systemgröße global für alle Partituren einer Datei (bzw. einer \book-Umgebung) zu verändern, wird set-global-staff-size benutzt:

```
#(set-global-staff-size 14)
```

Hiermit wird die Standardhöhe der Notensysteme auf 14 pt gesetzt. Die Schriftarten werden entsprechend verkleinert.

Um die Systemhöhe für jede Partitur einzeln zu verändern, muss

```
\score{
  ... 
  \layout {
    #(layout-set-staff-size 15)
  }
}
```

eingesetzt werden.

Die Feta-Glyphe stellt die Noten- und Musiksymbole für acht verschiedene Größen zur Verfügung. Jede Schriftgröße ist einer bestimmten Systemgröße angepasst: für kleinere Schriftgrößen werden die Zeichen etwas schwerer, um mit den ebenfalls dickeren Notenlinien zu harmonieren. Die empfohlenen Notensystemgrößen sind in der Tabelle aufgeführt:

<table>
<thead>
<tr>
<th>Schriftbezeichnung des Systems (pt)</th>
<th>Höhe des Systems (mm)</th>
<th>Benutzung</th>
</tr>
</thead>
<tbody>
<tr>
<td>feta11</td>
<td>11.22</td>
<td>3.9</td>
</tr>
<tr>
<td>feta13</td>
<td>12.60</td>
<td>4.4</td>
</tr>
<tr>
<td>feta14</td>
<td>14.14</td>
<td>5.0</td>
</tr>
<tr>
<td>feta16</td>
<td>15.87</td>
<td>5.6</td>
</tr>
<tr>
<td>feta18</td>
<td>17.82</td>
<td>6.3</td>
</tr>
<tr>
<td>feta20</td>
<td>20</td>
<td>7.0</td>
</tr>
<tr>
<td>feta23</td>
<td>22.45</td>
<td>7.9</td>
</tr>
<tr>
<td>feta26</td>
<td>25.2</td>
<td>8.9</td>
</tr>
</tbody>
</table>

Siehe auch
Notationsreferenz: [Auswahl der Notations-Schriftgröße], Seite 204.
Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.3 Umbrüche

4.3.1 Zeilenumbrüche

Zeilenumbrüche werden normalerweise automatisch erstellt. Sie werden so ausgewählt, dass die Zeilen weder gedrängt noch zu weit gespreizt wirken und aufeinander folgende Seiten einen ähnlichen Grauwert haben.

Einen manuellen Zeilenumbruch fügt man mit dem Befehl \break ein:

\relative c'' {
  c4 c c c | \break
c4 c c c |
}

\new Voice \with {
  \remove "Forbid_line_break_engraver"
}
Kapitel 4: Abstände

Genauso werden normalerweise Zeilenumbrüche auch verhindert, wenn Balken über die Taktenden hinausragen. Dieses Verhalten kann verändert werden, indem man \override Beam.breakable = ##t einstellt:

\relative c'' {
\override Beam.breakable = ##t
  c2. c8[ c | \break
c8 c] c2. |
}

Mit dem Befehl \noBreak wird ein Zeilenumbruch an dem entsprechenden Taktstrich verboten.

Die grundlegenden Einstellungen, die Einfluss auf die Zeilenlänge haben, sind indent (Einzug) und line-width (Zeilenbreite). Sie werden in der \layout-Umgebung eingestellt. Der erste Befehl bestimmt den Einzug der ersten Zeile, der zweite die Zeilenlänge der weiteren Notenzeilen.

Wenn ragged-right eingestellt ist (also in der \layout-Umgebung auf den Wert #t gesetzt wurde), werden die Systeme linksbündig gesetzt und nicht bis zum rechten Rand hin durchgezogen, sondern den Noten entsprechend gesetzt. Das ist oftmals nützlich für kleine Notenfragmente und um zu überprüfen, wie eng die Noten natürlicherweise gesetzt werden würden.

Die Option ragged-last verhält sich ähnlich zu ragged-right, aber wirkt sich nur auf die letzte Zeile eines Stücks aus.

\layout {
  indent = #0
  line-width = #150\mm
  ragged-last = ##t
}
Um Zeilenumbrüche zu erzwingen, die in festgelegten Intervallen stattfinden, kann der Befehl \break in Kombination mit unsichtbaren Noten und einer Wiederholung (\repeat) eingesetzt werden. Das folgende Beispiel etwa setzt die nächsten 28 Takte (im 4/4-Takt) in Zeilen zu jeweils 4 Takten (die auch nur hier umgebrochen werden):

```
<<
\repeat unfold 7 {
 s1 \noBreak s1 \noBreak
 s1 \noBreak s1 \break
 }
 { Hier die Noten... }
>>
```


**Vordefinierte Befehle**

\break, \noBreak.

**Siehe auch**

- Schnipsel: Abschnitt “Spacing” in Schnipsel.

### 4.3.2 Seitenumbrüche

Die Standardseitenumbrüche können verändert werden, indem man die Befehle \pageBreak bzw. \noPageBreak benutzt. Sie verhalten sich analog zu den Befehlen \break und \noBreak. Sie sollten an einem Taktstrich notiert werden. Diese Befehle erzwingen bzw. verbieten einen Seitenumbruch. Mit dem \pageBreak-Befehl wird natürlich gleichzeitig auch ein Zeilenumbruch erzwungen.

Die \pageBreak und \noPageBreak-Befehle können auch auf der höchsten Ebene einer Datei benutzt werden, etwa zwischen Partituren und Textbeschriftungen.

Es gibt auch vertikale Gegenstücke zu den Variablen \textit{\texttt{ragged-right}}, \textit{\texttt{ragged-bottom}} und \textit{\texttt{ragged-last-bottom}}. Wenn diese Variablen auf \texttt{#t} (wahr) gesetzt werden, werden im ersten Fall die Notensysteme auf allen Seiten eng nach oben orientiert gesetzt werden. Im zweiten Fall bezieht sich dies nur auf die letzte Seite. Zu Einzelheiten siehe Abschnitt 4.1.3 [Vertikale \textit{\texttt{\paper}}-Variablen mit festen Abständen], Seite 500.


```
\paper{
 #{define page-breaking ly:page-turn-breaking}
}
```


\bookpart {
Vordefinierte Befehle
\pageBreak, \noPageBreak.

Siehe auch
Notationsreferenz: \paper-Variablen für den Seitenumbruch], Seite 506.
Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.3.3 Optimale Seitenumbrüche

Siehe auch
Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.3.4 Optimale Umbrüche zum Blättern
Es ist oft nötig, die Seiten so umzubrechen, dass sich eine Pause am Ende jeder zweiten Seite befindet, damit der Musiker es leichter hat, die Seite umzublättern ohne das Spielen zu unterbrechen. Die ly:page-turn-breaking-Funktion versucht, Seitenumbrüche zu finden, die das Stauchen oder Strecken von Zeilen minimieren und gleichzeitig auch noch Seitenumbrüchen an angegebenen Stellen den Vorrang zu geben.

Die Funktion wird in zwei Schritten eingesetzt. Zunächst muss sie in der \paper-Umgebung aktiviert werden, wie gezeigt in Abschnitt 4.3.2 [Seitenumbrüche], Seite 514. Dann muss noch angegeben werden, welche Stellen bevorzugt für Seitenumbrüche benutzt werden sollen.

Für diesen zweiten Schritt gibt es zwei Methoden. Am Einfachsten ist es, die möglichen Seitenumbrüche mit dem Befehl \allowPageTurn an jeder Stelle manuell anzugeben.

Wenn Ihnen das zu aufwändig ist, können Sie den Page_turn_engraver zu einem Staff- oder Voice-Kontext hinzufügen. Dieser Engraver durchsucht den entsprechenden Kontext nach
Stellen ohne Noten. (Es wird also nicht nach Pausen gesucht, sondern nach Stellen ohne Noten. Dieses Verhalten verhindert, dass an polyphonen Stellen umgebrochen wird, wo nur in einer Stimme Pausen vorhanden sind.) Wenn eine derartige Stelle ohne Noten gefunden wird, fügt der Engraver den Befehl \allowPageTurn am letzten Taktstrich des Abschnitts ein. Wenn in dem Abschnitt ein besonderer Taktstrich vorkommt (wie etwa ein Doppelstrich), wird der Befehl nach diesem Taktstrich gesetzt.

Der Page_turn_engraver liest die Kontexteigenschaft minimumPageTurnLength um zu erkennen, wie lang eine Stelle frei von Noten sein muss, damit ein Seitenumbruch in Frage kommt. Der Standardwert hierfür ist (ly:make-moment 1/1). Wenn Sie Seitenumbrüche zum Umblättern ausschalten wollen, können Sie einen sehr großen Wert angeben.

\new Staff \with { \consists "Page_turn_engraver" } 
{ 
a4 b c d | 
R1 | % Ein Seitenumbruch zum Umblättern erlaubt 
a4 b c d | 
\set Staff.minimumPageTurnLength = #(ly:make-moment 5/2) 
R1 | % Seitenumbruch nicht erlaubt 
a4 b r2 | 
R1*2 | % Seitenumbruch erlaubt 
a1 
}

Der Page_turn_engraver erkennt Wiederholungen vom Typ volta. Ein Seitenumbruch zum Umblättern wird nur zugelassen, wenn vor und nach der Wiederholung genug Zeit ist, um die Seite wieder zurückzublättern. Wenn die Wiederholung sehr kurz ist, kann auch Umblättern verboten werden. Wenn Sie die Kontexteigenschaft minimumRepeatLengthForPageTurn definieren, erlaubt der Page_turn_engraver nur Umblättern in Wiederholungen, deren Dauer länger als dieser Wert ist.

Die Seitenumblätter-Befehle \pageTurn, \noPageTurn und \allowPageTurn können auch auf oberster Dateiebene benutzt werden, etwa zwischen Partituren und Textabschnitten.

Vordefinierte Befehle
\pageTurn, \noPageTurn, \allowPageTurn.

Siehe auch
Schnipsel: Abschnitt “Spacing” in Schnipsel.

Bekannte Probleme und Warnungen
In einer Partitur sollte nur ein Page_turn_engraver vorkommen. Wenn mehr als einer definiert werden, stören sie sich gegenseitig.

4.3.5 Minimale Seitenumbrüche
Die ly: minimal-breaking-Funktion benötigt nur minimale Berechnungen, um die Seitenumbrüche zu bestimmen. Die Seite wird mit möglichst vielen Systemen gefüllt und dann zur nächsten Seite gewechselt. Die Funktion kann benutzt werden um Partituren mit vielen Seiten zu setzen, wenn die anderen Seitenumbruchsfunktionen zu langsam wären oder zu viel Speicher beanspruchten. Auch für Seiten mit viel Text ist die Funktion geeignet. Sie wird folgendermaßen aktiviert:

\paper { 
  page-breaking = #ly: minimal-breaking 
}
Siehe auch
Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.3.6 Eine-Seite-Seitenumbrüche

Die Seitenbreite wird angepasst, sodass die längste Partitur auf eine Zeile passt. Die Variablen paper-width, line-width und indent in der \paper-Umgebung werden ignoriert, wenn auch left-margin und right-margin noch beachtet werden. Die Höhe der Seite wird nicht verändert.

4.3.7 Ausdrückliche Umbrüche
Es kann vorkommen, dass LilyPond direkte \break oder \pageBreak-Befehl nicht beachtet. Mit folgenden Einstellungen kann dieses Verhalten ausgeschaltet werden:

\override NonMusicalPaperColumn.line-break-permission = ##f
\override NonMusicalPaperColumn.page-break-permission = ##f

Wenn line-break-permission die Einstellung falsch (##f) hat, werden Zeilenumbrüche nur an den Befehlen \break eingefügt und nirgendwo anders. Wenn page-break-permission die Einstellung falsch (##f) hat, werden Seitenumbrüche nur an den Befehlen \pageBreak eingefügt und nirgendwo anders.

\paper {
  indent = #0
  ragged-right = ##t
  ragged-bottom = ##t
}

\score {\relative {c''8 c c c}}

\layout {
  \context {\Score
    \override NonMusicalPaperColumn.line-break-permission = ##f
    \override NonMusicalPaperColumn.page-break-permission = ##f
  }
}
}
Siehe auch

Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.3.8 Eine zusätzliche Stimme für Umbrüche benutzen

Zeilen- und Seitenumbruchbefehle werden normalerweise direkt zusammen mit den Noten eingegeben.

\[ \text{music} = \text{relative} \{ \text{c''4 c c c} \} \]
\[
\text{\score{ }
\text{new Staff{ }
\text{repeat unfold 2{ \text{music} } \text{break} }
\text{repeat unfold 3{ \text{music} } }
\text{}}}
\]

Hierdurch sind zwar die Befehle \text{break} und \text{pageBreak} einfach zu notieren, es werden aber Informationen zur Notation mit Informationen zur Anordnung auf der Seite vermischt. Man kann diese Informationen auch voneinander trennen, indem man eine zusätzliche Stimme einfügt, in
der Zeilen- und Seitenumbrüche vorgenommen werden. Diese zusätzliche Stimme enthält nur unsichtbare Noten und die Umbruchbefehle:

\begin{verbatim}
\score {
    \new Staff <<
        \new Voice {
            s1 * 2 \break
            s1 * 3 \break
            s1 * 6 \break
            s1 * 5 \break
        }
        \new Voice {
            \repeat unfold 2 { \music }
            \repeat unfold 3 { \music }
            \repeat unfold 6 { \music }
            \repeat unfold 5 { \music }
        }
    >>
}
\end{verbatim}


\begin{verbatim}
\score {
    \new Staff <<
        \new Voice {
            \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details #'(Y-offset
            s1 * 2 \break
        }
    >>
}
\end{verbatim}
Kapitel 4: Abstände

4.4 Vertikale Abstände

Vertikale Abstände werden durch drei Eigenschaften bestimmt: wieviel Platz frei ist (etwa Papiergröße und Ränder), wieviel Platz zwischen Systemgruppen (engl. system) gesetzt werden soll und wieviel Platz zwischen Notensystemen (engl. staff, Pl. staves) innerhalb von Gruppen gesetzt wird.

4.4.1 Flexible vertikale Abstände in Systemgruppen

Drei unterschiedliche Mechanismen kontrollieren das flexible Abstandaufteilung in Systemgruppen, einer für jede der folgenden Kategorien:

- ungruppierte Systeme,
- Systemgruppen (Systeme innerhalb einer staff-group wie etwa ChoirStaff usw.) und

Siehe auch

Notationsreferenz: Abschnitt 4.4 [Vertikale Abstände], Seite 520.
Schnipsel: Abschnitt “Spacing” in Schnipsel.
Nicht-Notensystemzeilen (wie etwa Lyrics (Gesangstext), ChordNames (Akkordbezeichnungen) usw.).


Es ist zu beachten, dass der Platzverteilungsmechanismus, der in diesem Abschnitt behandelt wird, nur die vertikale Platzierung von Systemen und nicht-Systemzeilen in einzelnen Systemgruppen behandelt. Die vertikale Platzierung zwischen einzelnen Systemgruppen, Partituren, Beschriftungen usw. und den Rändern wird durch \paper-Variablen kontrolliert, die Abschnitt 4.1.4 [Vertikale \paper-Variablen mit flexiblen Abständen], Seite 501.

Eigenschaften für Abstände innerhalb von Systemgruppen


Jeder Abstand wird zwischen den \textit{Referenzpunkten} der zwei Objekten gemessen. Der Referenzpunkt eines Notensystems ist die vertikale Mitte seines StaffSymbol-Objekts (also die Mittellinie, wenn line-count (Notenlinienzähler) ungrade ist, oder der mittlere Zwischenraum, wenn line-count grade ist). Die Referenzpunkte für einzelne Nicht-Notensystemzeilen ergibt sich aus der folgenden Tabelle:

<table>
<thead>
<tr>
<th>Nicht-Notensystemzeile</th>
<th>Referenzpunkt</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChordNames</td>
<td>Grundlinie</td>
</tr>
<tr>
<td>NoteNames</td>
<td>Grundlinie</td>
</tr>
<tr>
<td>Lyrics</td>
<td>Grundlinie</td>
</tr>
<tr>
<td>Dynamics</td>
<td>vertikale Mitte</td>
</tr>
<tr>
<td>FiguredBass</td>
<td>höchster Punkt</td>
</tr>
<tr>
<td>FretBoards</td>
<td>Oberlinie</td>
</tr>
</tbody>
</table>

Im nächsten Bild zeigen horizontale Striche die Positionen dieser Referenzpunkte an:

- baseline: \texttt{ChordNames NoteNames Lyrics}
- vertical center: \texttt{Dynamics mp fp}
- highest point: \texttt{FiguredBass 5}
Jeder der vertikalen Platzierungs-Grobeigenschaften (außer staff-affinity) benutzt die gleiche Alistenstruktur wie die \paper- Variablen, behandelt in Abschnitt 4.1.4 [Vertikale \paper-Variablen mit flexiblen Abständen], Seite 501. Besondere Methoden um Alisten zu verändern finden sich in Abschnitt 5.3.6 [Alisten verändern], Seite 578. Grob-Eigenschaften sollten mit dem \override-Befehle innerhalb einer \score- oder \layout-Umgebung angepasst werden, nicht innerhalb einer \paper-Umgebung.

Das folgende Beispiel zeigt die beiden Arten, Alisten zu modifizieren. Der erste Aufruf verändert nur einen Schlüsselwert einzeln, während der zweite die Eigenschaft komplett neu definiert:

\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing.basic-distance = #10
} { ... }

\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing =
    #'((basic-distance . 10)
    (minimum-distance . 9)
    (padding . 1)
    (stretchability . 10))
} { ... }

Um Platzierungseinstellungen global vorzunehmen, müssen sie in der \layout-Umgebung vorgenommen werden:

\layout {
  \context {
    \Staff
    \override VerticalAxisGroup.default-staff-staff-spacing.basic-distance = #10
  }
}


**Eigenschaften des VerticalAxisGroup-Grobs**

VerticalAxisGroup-Eigenschaften werden normalerweise mit einem \override-Befehl auf Staff-(Notensystem-)Ebene (oder entsprechend) vorgenommen.

staff-staff-spacing

System-System-Platzierung

Wird benutzt, um den Abstand zwischen dem aktuellen Notensystem und dem Notensystem direkt darunter in der gleichen Notensystemgruppe zu bestimmen, auch wenn eine oder mehrere Nicht-Notensystemzeilen (wie etwa Lyrics) dazwischen stehen. Gilt nicht für das unterste System einer Systemgruppe.

Die Eigenschaft staff-staff-spacing einer VerticalAxisGroup ist eine Scheme-Funktion, welche die Eigenschaften vom StaffGrouper-Grob den Notensystemen zuweist, wenn sie einer Systemgruppe angehören, bzw. die default-staff-staff-spacing-Eigenschaft bei einem einzelnen Notensystem. Dadurch können Systeme unterschiedlich in Abhängigkeit von ihrer Zugehörigkeit
zu einer Gruppe (StaffGroup) platziert werden. Wenn gleichmäßige Aufteilung ohne Berücksichtigung von Gruppierungen gewünscht ist, kann diese Funktion durch eine flexible Platzierungs-Aliste ersetzt werden, wobei man die vollständige Redefinition auf die oben gezeigte Weise vornehmen muss.

`default-staff-staff-spacing`

Normale-System-System-Platzierung
Eine flexible Platzierungs-Aliste, die den Wert von `staff-staff-spacing` für ungruppierte Notensysteme einstellt, es sei denn, `staff-staff-spacing` werde explizit mit \override eingestellt.

`staff-affinity`

System-Anziehung

`nonstaff-relatedstaff-spacing`

Nicht-Notensystem-verwandtesSystem-Platzierung

`nonstaff-nonstaff-spacing`

Nicht-Notensystemzeile-Nicht-Notensystemzeile-Platzierung
Der Abstand zwischen der aktuellen Nicht-Notensystemzeile und der nächsten Nicht-Notensystemzeile in der Richtung von staff-affinity, wenn beide sich auf der gleichen Seite des verwandten Notensystems befinden und staff-affinity entweder UP oder DOWN ist.

`nonstaff-unrelatedstaff-spacing`

Nicht-Notensystemzeile-Nicht-verwandtesSystem-Platzierung
Der Abstand zwischen der aktuellen Nicht-Notensystemzeile und dem Notensystem in der gegenüberliegenden Richtung von staff-affinity, wenn keine anderen Nicht-Notensystemzeilen dazwischen auftreten und staff-affinity entweder UP
oder DOWN ist. Das kann benutzt werden, um einen Minimalfüllabstand (padding) zwischen einer Lyrics-Gesangstextzeile und dem zugehörigen Notensystem zu verlängern.

**Eigenschaften des StaffGrouper-Grobs**

*StaffGrouper*-Eigenschaften werden normalerweise mit einem `\override`-Befehl auf *StaffGroup*-Ebene (oder entsprechend) eingestellt.

**staff-staff-spacing**

Notensystem-Notensystem-Abstand

Der Abstand zwischen zwei aufeinanderfolgenden Notensystemen in der aktuellen *StaffGroup*. Die `staff-staff-spacing`-Eigenschaft des *VerticalAxisGroup*-Grobs eines einzelnen Notensystems kann mit `\override` in andere Platzierungswerte für dieses Notensystem geändert werden.

**staffgroup-staff-spacing**

Systemgruppe-System-Abstand


Siehe auch

Notationsreferenz: Abschnitt 4.1.4 [Vertikale \paper-Variablen mit flexiblen Abständen], Seite 501, Abschnitt 5.3.6 [Listen verändern], Seite 578.


**Abstände von nicht gruppierten Notensystemen**

*Notensysteme* (wie etwa *Staff, DrumStaff, TabStaff* usw.) sind Kontexte, die eine oder mehrere Stimmen-Kontexte enthalten, aber keine anderen Notensysteme enthalten können.

Folgende Eigenschaften beeinflussen die Abstände von *nicht gruppierten* Notensystemen:

- *VerticalAxisGroup*-Eigenschaften:
  - `default-staff-staff-spacing`
  - `staff-staff-spacing`

Diese Eigenschaften sind einzeln oben behandelt worden, siehe [Eigenschaften für Abstände innerhalb von Systemgruppen], Seite 521.

Zusätzliche Eigenschaften kommen hinzu für Notensysteme, die Teil einer Gruppierung (*StaffGroup*) werden, siehe [Abstände von gruppierten Notensystemen], Seite 525.

Folgendes Beispiel zeigt, wie die `default-staff-staff-spacing`-Eigenschaft sich auf die Platzierung von nicht-gruppierten Notensystemen auswirken kann. Wenn man die gleichen `\override`-Befehle auf `staff-staff-spacing` anwendet, ergäbe das den selben Effekt, würde sich aber auch in Fällen auswirken, in denen die Systeme gruppiert sind.

```latex
\layout {
  \context {
```
Kapitel 4: Abstände

\Staff
\override VerticalAxisGroup.default-staff-staff-spacing = '#('(basic-distance . 8)
  (minimum-distance . 7)
  (padding . 1))
}
}

<<
% The very low note here needs more room than 'basic-distance
% can provide, so the distance between this staff and the next
% is determined by 'padding.
\new Staff { b,2 r | }

% Here, 'basic-distance provides enough room, and there is no
% need to compress the space (towards 'minimum-distance) to make
% room for anything else on the page, so the distance between
% this staff and the next is determined by 'basic-distance.
\new Staff { \clef bass g2 r | }

% By setting 'padding to a negative value, staves can be made to
% collide. The lowest acceptable value for 'basic-distance is 0.
\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing = '#'(basic-distance . 3.5)
  (padding . -10))
} { \clef bass g2 r | }
\new Staff { \clef bass g2 r | }
>>

Siehe auch

Installierte Dateien: scm/define-grobs.scm.
Schnipsel: Abschnitt “Spacing” in Schnipsel.

Abstände von gruppierten Notensystemen

In Orchesterpartituren und anderen großen Partituren werden Notensysteme normalerweise in
Gruppen zusammengefasst. Der Platz zwischen Gruppen ist normalerweise größer als der Zwischenraum zwischen einzelnen Notensystemen der gleichen Gruppe.

Gruppierte Notensysteme (wie StaffGroup, ChoirStaff, GrandStaff usw.) sind Kontexte, die mehr als ein Notensystem gleichzeitig enthalten können.
Folgende Eigenschaften beeinflussen die Platzierung von Notensystemen innerhalb von Gruppen:

- **VerticalAxisGroup-Eigenschaften:**
  - staff-staff-spacing

- **StaffGrouper-Eigenschaften:**
  - staff-staff-spacing
  - staffgroup-staff-spacing

Diese Grob-Eigenschaften sind weiter oben einzeln beschrieben, siehe [Eigenschaften für Abstände innerhalb von Systemgruppen], Seite 521.

Das folgende Beispiel zeigt, wie Eigenschaften des `StaffGrouper`-Grobs die Platzierung von gruppierten Notensystemen beeinflussen kann:

```latex
\layout {
  \context {
    \Score
    \override StaffGrouper.staff-staff-spacing.padding = #0
    \override StaffGrouper.staff-staff-spacing.basic-distance = #1
  }
}

<<
  \new PianoStaff \with {
    \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #20
  } <<
  \new Staff { c'1 }
  \new Staff { c'1 }
>>

\new StaffGroup <<
  \new Staff { c'1 }
  \new Staff { c'1 }
>>
```

Siehe auch

Installierte Dateien: `scm/define-grobs.scm`.
Schnipsel: Abschnitt “Spacing” in Schnipsel.


Abstände von nicht-Notensystemzeilen


Folgende Eigenschaften beeinflussen die Abstände von Nicht-Notensystemzeilen:

- *VerticalAxisGroup*-Eigenschaften:
  - *staff-affinity*
  - *nonstaff-relatedstaff-spacing*
  - *nonstaff-nonstaff-spacing*
  - *nonstaff-unrelatedstaff-spacing*

Diese Grob-Eigenschaften sind weiter oben einzeln beschrieben; siehe [Eigenschaften für Abstände innerhalb von Systemgruppen], Seite 521.

Das folgende Beispiel zeigt, wie die *nonstaff-nonstaff-spacing*-Eigenschaft die Platzierung von aufeinanderfolgenden Nicht-Notensystemzeilen beeinflussen kann. Indem hier der Wert von *stretchability* auf einen sehr hohen Wert gesetzt wird, kann der Gesangstext sehr viel weiter als normal gespreizt werden:

\layout {
\context {
  \Lyrics
  \override VerticalAxisGroup.nonstaff-nonstaff-spacing.stretchability = #1000
}
}

\new StaffGroup
<<
\new Staff \with {
  \override VerticalAxisGroup.staff-staff-spacing = #'((basic-distance . 30))
} { c'1 }
\new Lyrics \with {
  \override VerticalAxisGroup.staff-affinity = #UP
} \lyricmode { up }
\new Lyrics \with {
  \override VerticalAxisGroup.staff-affinity = #CENTER
} \lyricmode { center }
\new Lyrics \with {
  \override VerticalAxisGroup.staff-affinity = #DOWN
} \lyricmode { down }
\new Staff { c'1 }
>>
4.4.2 Explizite Positionierung von Systemen

Man kann die flexiblen Einstellungen der vertikalen Abstände, wie sie im vorigen Abschnitt erklärt wurden, als eine Sammlung verschiedener Einstellmöglichkeiten verstehen, die vor allem die Größe des vertikalen Platzes zwischen Notensystemen und Gruppen auf der Seite kontrollieren.


NonMusicalPaperColumn #'line-break-system-details akzeptiert eine Liste aus drei unterschiedlichen Einstellungen:

- X-offset
- Y-offset
- alignment-distances

Veränderungen von Grobs (wozu auch NonMusicalPaperColumn gehört), können an drei unterschiedlichen Stellen in der Quelldatei vorgenommen werden:

- mitten im Notentext
- in einer \context-Umgebung
- in einer \with-Umgebung

Wenn der Grob NonMusicalPaperColumn verändert werden soll, wird der \override-Befehl in der \context oder \with-Umgebung eingesetzt. Wenn die Veränderungen aber mitten im Notentext stattfinden sollen, müssen Sie den Befehl \overrideProperty einsetzen. Einige Beispiele für eine Veränderungen von NonMusicalPaperColumn mit dem \overrideProperty-Befehl sind hier aufgelistet:

\overrideProperty NonMusicalPaperColumn.line-break-system-details '((X-offset . 20))

\overrideProperty NonMusicalPaperColumn.line-break-system-details '((Y-offset . 40))

\overrideProperty NonMusicalPaperColumn.line-break-system-details '((X-offset . 20) (Y-offset . 40))
Um zu verstehen, wie jede dieser unterschiedlichen Einstellungen funktioniert, wollen wir uns ein Beispiel vornehmen, dass überhaupt keine Einstellungen (d.h. \override-Befehle) enthält:

\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
\score {
<<
\new Staff <<
\new Voice {
  s1*5 \break
  s1*5 \break
  s1*5 \break
}
\new Voice { \repeat unfold 15 { c'4 c' c' c' } }
>>
\new Staff {
  \repeat unfold 15 { d'4 d' d' d' }
}
>>
}
}

Diese Partitur nimmt Zeilen- und Seitenumbruchinformationen in einer eigenen Stimme vor. Mit dieser Methode kann die Layout-Information einfach von den Noten getrennt werden, was sehr hilfreich ist, wenn das Beispiel komplizierter wird. Siehe auch Abschnitt 4.3.8 [Eine zusätzliche Stimme für Umbrüche benutzen], Seite 518.

Ausdrückliche \break-Befehle teilen die Noten in sechs Takte lange Zeilen. Die vertikale Platzverteilung wird von LilyPond errechnet. Um den vertikalen Beginn einer jeden System-
Kapitel 4: Abstände

gruppe genau anzugeben, kann \texttt{Y-offset} in der \texttt{line-break-system-details}-Eigenschaft des \texttt{NonMusicalPaperColumn}-Grobs wie in dem Beispiel ersichtlich benutzt werden:

\begin{verbatim}
\header { tagline = \##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    \new Staff <<
    \new Voice {
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details #'((Y-offset s1*5 \break
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details #'((Y-offset s1*5 \break
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details #'((Y-offset s1*5 \break
      }\new Voice { \repeat unfold 15 { c'4 c' c' c' } }
    }\new Staff {
      \repeat unfold 15 { d'4 d' d' d' } 
    }
  }
  }
}\end{verbatim}
In der line-break-system-details-Eigenschaft kann eine Liste mit vielen Einstellungen eingegeben werden, aber hier wird nur eine Einstellung angegeben. Die Y-offset-Eigenschaft bestimmt hier die exakte vertikale Position auf der Seite, an welcher jede neue Systemgruppe begonnen wird.


\header { tagline = ##f } \
\paper { left-margin = 0\mm } \
book { \
  \score { \
    << \
      \new Staff << \
      \new Voice { \
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details #'((Y-offset \
          (alignment-distances . (15))) \
          s1*5 \break \
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details #'((Y-offset \
          (alignment-distances . (15))) \
          s1*5 \break \
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details #'((Y-offset \
          (alignment-distances . (15))) \
          s1*5 \break 
      } 
    } 
  }
Kapitel 4: Abstände

Dem \texttt{line-break-system-details}-Attribut des \texttt{NonMusicalPaperColumn}-Grobs werden zwei unterschiedliche Eigenschaften zugewiesen. Auch wenn die Aliste der Attribute von \texttt{line-break-system-details} sehr viel mehr Platzierungsparameter akzeptiert (wie etwa ein korrespondierendes \texttt{X-offset}-Paar), müssen hier nur die Parameter \texttt{Y-offset} und \texttt{alignment-distances} gesetzt werden, um den vertikalen Beginn jedes Systems und jeder Systemgruppe zu kontrollieren. \texttt{Y-offset} bestimmt also die vertikale Position von Systemgruppen und \texttt{alignment-distances} die vertikale Position von einzelnen Notensystemen.
\score {
  \new Staff <<
  \new Voice {
    \overrideScore.NonMusicalPaperColumn.line-break-system-details #'(\(alignment-distances . (30 10)))
    s1*5 \break
    \overrideScore.NonMusicalPaperColumn.line-break-system-details #'(\(alignment-distances . (10 10)))
    s1*5 \break
    \overrideScore.NonMusicalPaperColumn.line-break-system-details #'(\(alignment-distances . (10 30)))
    s1*5 \break
  }
  \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
  >>
  \new StaffGroup <<
  \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
  \new Staff { \repeat unfold 15 { e'4 e' e' e' } }
  >>
  >>
}
}
Einige Dinge sollten beachtet werden:

- Wenn `alignment-distances` benutzt wird, werden Gesangstextzeilen nicht als ein System gezählt.
- Weil die Einstellungen von `NonMusicalPaperColumn #'line-break-system-details` es möglich machen, Notensysteme und Gruppen an beliebigen Stellen auf der Seite
zu platzieren, kann man damit auch Ränder überschreiben oder sogar Notensysteme übereinander platzieren. Sinnvolle Werte für diese Parameter werden derartiges Verhalten vermeiden.

Siehe auch

Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.4.3 Vermeidung von vertikalen Zusammenstößen

Intuitiv gibt es in der Notation einige Objekte, die zu dem Notensystem gehören, und einige andere, die immer außerhalb des Notensystems positioniert werden sollten. Zu diesen letzteren gehören etwa Übungszeichen, Textbeschriftung und Dynamikbezeichnung (die als Objekte außerhalb des Systems bezeichnet werden können). LilyPonds Regeln um diese Objekte zu positionieren lautet: so nah am Notensystem wie möglich, aber gerade so weit weg, dass sie nicht mit anderen Objekten zusammenstoßen.


Zuerst werden alle Objekte gesetzt, die nicht außerhalb des Systems gehören. Dann werden die Objekte außerhalb des Systems nach dem Wert ihrer outside-staff-priority (in aufsteigender Anordnung) sortiert. Eins nach dem anderen werden diese Objekte schließlich genommen und so platziert, dass sie nicht mit den Objekten zusammenstoßen, die bereits platziert worden sind. Wenn also zwei Objekte außerhalb des Systems um den gleichen Platz streiten, wird das mit dem geringeren Wert von outside-staff-priority näher an das entsprechende Notensystem gesetzt.

\relative c'' { c4_"Text"\pp r2. \once \override TextScript.outside-staff-priority = #1 c4_"Text"\pp % this time the text will be closer to the staff r2. % by setting outside-staff-priority to a non-number, % we disable the automatic collision avoidance \once \override TextScript.outside-staff-priority = ##f \once \override DynamicLineSpanner.outside-staff-priority = ##f c4_"Text"\pp % now they will collide }

Der Platz, der zwischen einem Objekt außerhalb des Systems und dem vorhergehenden Objekt eingefügt werden kann (auch als padding bezeichnet), kann durch outside-staff-padding kontrolliert werden.

\once \override TextScript.outside-staff-padding = #0 a'"This text is placed very close to the note" \once \override TextScript.outside-staff-padding = #3

Siehe auch

Schnipsel: Abschnitt “Spacing” in *Schnipsel*.

### 4.5 Horizontale Abstände

#### 4.5.1 Überblick über horizontale Abstände


```
c2 c4. c8 c4. c8 c4. c8 c8
```
Normalerweise ist \textit{spacing-increment} definiert als 1.2 mal der Abstand zwischen zwei Notenlinien, was in etwa die Breite eines Notenkopfes ist. \textit{shortest-duration-space} ist definiert als 2.0, was bedeutet, dass die kürzeste Note 2.4 Notenlinienabstände 2.0 mal der Wert von \textit{spacing-increment}) horizontalen Abstand erhält. Der Abstand wird von der linken Kante des Symbols errechnet, so dass die kürzeste Note üblicherweise von 1 NKA Abstand gefolgt wird.

Wenn diese Herangehensweise konsequent angewandt würde, würde eine einzige Zweiunddreiβigstel eine Partitur, in der vor allem Achtel und Sechzehntel vorkommen, sehr weit auseinanderdehnen. Die kürzeste Note wäre nun keine Sechzehntel mehr, sondern eine Zweiunddreißigstel, wodurch an jede Note der Wert von 1 NKA hinzugefügt würde. Um das zu vermeiden, ist die kürzeste Dauer für die Platzverteilung nicht die kürzeste Note einer Partitur, sondern die, die am häufigsten vorkommt.

Die Notendauer, die am häuβigsten vorkommt, wird auf folgende Weise bestimmt: in jedem Takt wird die kürzeste Note bestimmt. Die häufigste kürzeste Note wird dann als Grundlage für die Platzverteilung der Noten herangezogen, mit der Bedingung, dass diese kürzeste Note immer ein Achtel oder kürzer sein soll. Die kürzeste Dauer wird ausgegeben, wenn \texttt{lilypond} mit der Option \texttt{--verbose} aufgerufen wird.

Diese Dauern können aber auch angepasst werden. Wenn Sie die Eigenschaft \textit{common-shortest-duration} in dem \texttt{SpacingSpanner} setzen, dann wird hiermit die Grunddauer für die Platzverteilung eingestellt. Die maximale Dauer für diesen Grundwert (normalerweise eine Achtel) wird definiert mit \texttt{base-shortest-duration}.

Noten, die noch kürzer sind als die häufigste kürzeste Note, werden durch einen Platz voneinander getrennt, der proportional zu ihrer Dauer in Beziehung zur häufigsten kürzesten Note ist. Wenn also nur ein paar Sechzehntel zu dem obigen Beispiel hinzugefügt werden, würden sie von 1/2 NKA gefolgt werden:

\begin{verbatim}
c2 c4 c8 c4 c16 [ c] c4 c8 c8 c4 c4 c4
\end{verbatim}

In dem \textit{Aufsatz zum automatisierten Notensatz} wurde erklärt, dass die Richtung der Notenähse die Platzverteilung beeinflusst (siehe Abschnitt “Optischer Ausgleich” in \textit{Aufsatz}). Das wird kontrolliert durch die \texttt{stem-spacing-correction}-Eigenschaft in dem \texttt{NoteSpacing}-Objekt. Dieses Objekt wird für jeden \texttt{Voice}-Kontext erstellt. Das \texttt{StaffSpacing}-Objekt (in einem \texttt{Staff}-Kontext erstellt) enthält die gleiche Eigenschaft, um die Verteilung von Hälsen neben Taktilien zu kontrollieren. In dem folgenden Beispiel werden diese Einstellungen gezeigt, einmal mit den Standardwerten und dann mit größeren Werten, damit man sie besser sieht:

\begin{verbatim}
c\end{verbatim}

Proportionale Notation ist unterstützt, siehe Abschnitt 4.5.5 [Proportionale Notation], Seite 541.
Siehe auch

Schnipsel: Abschnitt “Spacing” in Schnipsel.
Referenz der Interna: Abschnitt “SpacingSpanner” in Referenz der Interna, Abschnitt “NoteSpacing” in Referenz der Interna, Abschnitt “StaffSpacing” in Referenz der Interna, Abschnitt “NonMusicalPaperColumn” in Referenz der Interna.

Bekannte Probleme und Warnungen

Es gibt keine sinnvolle Möglichkeit, die horizontale Verteilung der Noten zu unterdrücken. Die folgende Problemumgehung, mit der dehnbare Abstände (padding) eingesetzt werden, kann benutzt werden, um zusätzlichen Platz in eine Partitur einzufügen.

\override Score.NonMusicalPaperColumn.padding = #10

Es gibt derzeit keine Möglichkeit, den Platz zu verringern.

4.5.2 Eine neuer Bereich mit anderen Abständen

Neue Abschnitte mit unterschiedlichen Notenabstandsparametern können mit dem Befehl newSpacingSection begonnen werden. Das ist hilfreich, wenn in verschiedenen Abschnitten die Verhältnisse von kurzen und langen Noten sehr unterschiedlich ausfallen.

Im folgenden Beispiel wird durch die neue Taktart ein neuer Abschnitt begonnen, in dem die Sechzehntel weiter außen liegen gesetzt werden sollen.

\time 2/4
\newSpacingSection
\time 4/16
\score {
  \relative {
    g'4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e e g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
  }
}

Der newSpacingSection-Befehl erstellt ein neues SpacingSpanner-Objekt, weshalb auch neue Anpassungen mit dem \override-Befehl an dieser Stelle eingesetzt werden können.

Siehe auch

Schnipsel: Abschnitt “Spacing” in Schnipsel.
Referenz der Interna: Abschnitt “SpacingSpanner” in Referenz der Interna.

4.5.3 Horizontale Abstände verändern


\score {
  \relative {
    g'4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e e g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
  }
}
Ausgewählte Schnipsel

Standardmäßig wird die Platzverteilung in Triolen und andern rhythmischen Aufteilungen nach verschiedenen nicht von der Dauer abgeleiteten Faktoren (wie Versetzungssymbolen, Schlüsselwechseln usw.) berechnet. Um diese Symbole zu ignorieren und eine gleichmäßige Verteilung der Noten zu erzwingen, kann die gleichmäßige Dehnung (engl. uniform stretching) zu Beginn einer Partitur mit \texttt{Score.SpacingSpanner \# 'uniform-stretching} eingeschaltet werden:

\begin{verbatim}
\score {
  \new Staff {
    \tuplet 5/4 {
      c8 c8 c8 c8 c8
    }
    c8 c8 c8 c8
  }
  \new Staff {
    c8 c8 c8 c8
    \tuplet 5/4 {
      c8 c8 c8 c8 c8
    }
  }
  \layout {
    \context {\Score
      \override SpacingSpanner.uniform-stretching = ##t
    }
  }
}
\end{verbatim}

Wenn \texttt{strict-note-spacing} eingestellt ist, werden Noten gesetzt, ohne dass Schlüssel, Takttlinie oder Verzierungsnoten zusätzlichen Platz erhalten.

\begin{verbatim}
\override Score.SpacingSpanner.strict-note-spacing = ##t
\new Staff { c8[ c \clef alto c \grace { c16 c } c8 c c] c32[ c32 ]}
\end{verbatim}

Siehe auch

Schnipsel: Abschnitt „Spacing“ in \texttt{Schnipsel}. 
4.5.4 Zeilenlänge

Die grundlegenden Einstellungen, die Einfluss auf die Zeilenlänge haben, sind `indent` (Einzug) und `line-width` (Zeilenbreite). Sie werden in der `\layout`-Umgebung eingestellt. Der erste Befehl bestimmt den Einzug der ersten Zeile, der zweite die Zeilenlänge der weiteren Notenzeilen.

Wenn `ragged-right` eingestellt ist (als in der `\layout`-Umgebung auf den Wert `#t` gesetzt wurde), werden die Systeme linksbündig gesetzt und nicht bis zum rechten Rand hin durchgezogen, sondern den Noten entsprechend gesetzt. Das ist oft nützlich für kleine Notenfragmente und um zu überprüfen, wie eng die Noten natürlich gesetzt werden würden. Die normale Einstellung ist unwahr (`#f`), aber wenn eine Partitur nur aus einer Zeile besteht, ist der Standardwert wahr.

Die Option `ragged-last` verhält sich ähnlich zu `ragged-right`, aber wirkt sich nur auf die letzte Zeile eines Stücks aus. Für diese letzte Zeile gibt es keine Einschränkungen. Das Resultat erinnert an Textabsätze im Blocksatz, wo die letzte Zeile des Absatzes mit ihrer natürlichen Länge gesetzt wird.

\layout {
  indent = #0
  line-width = #150
  ragged-last = ##t
}

Siehe auch

Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.5.5 Proportionale Notation


LilyPond hat Unterstützung für fünf verschiedene Einstellungen der proportionalen Notation, die alle zusammen oder jede für sich benutzt werden können:

- `proportionalNotationDuration` (proportionale Notendauer)
- `uniform-stretching` (gleichmäßige Dehnung)
- `strict-note-spacing` (strenge Notenverteilung)
- `\remove "Separating_line_group_engraver"` (entferne Liniengruppentrennungsentgraver)
- `\override PaperColumn.used = ##t` (PapierSpalte benutzt = wahr)

In den Beispielen unten werden diese fünf unterschiedlichen Einstellungen für die proportionale Notation vorgestellt und ihre Wirkungen untereinander illustriert.

Es soll mit diesem 1 Takt langen Beispiel begonnen werden, in welchem die klassischen Abstände und Flattersatz (ragged-right) eingesetzt werden:

```
\score {
  \new RhythmicStaff { 
    c'2 
    c'16 c'16 c'16 c'16 
  \tuplet 5/4 { 
```
Die Halbe, mit der der Takt beginnt, braucht weitaus weniger Platz als die Hälfte des Taktes. Gleichermaßen haben die Sechzehntel und die Sechzehntel-Quintolen (oder Zwanzigstel), mit denen der Takt endet, insgesamt weitaus mehr als die Hälfte der Taktbreite.

Im klassischen Notensatz kann dieses Verhalten genau das gewünschte Ergebnis bringen, weil dadurch horizontaler Platz von der Halben weggenommen werden kann und so insgesamt Platz in dem Takt eingespart wird.

Wenn allerdings ein Zeitstrahl oder andere zeitliche ablaufende Graphiken über oder unter dem Takt eingefügt werden soll, braucht man eine Notenplatzierung, die exakt der von ihnen eingenommenen Dauer entspricht. Auf folgende Art wird die proportionale Notation eingeschaltet:

```ly
\score {
  \context {
    \Score
    proportionalNotationDuration = #(ly:make-moment 1/20)
  }
}
```

Die Eigenschaft `proportionalNotationDuration` braucht ein Argument, welches die Referenzdauer ist, anhand welcher alle Noten platziert werden. Hier wird die LilyPond Scheme-Funktion `make-moment` eingesetzt. Sie braucht zwei Argumente: einen Zähler und einen Nenner,


\score {
  \new RhythmicStaff {
    c'2
c'16 c'16 c'16 c'16
  \tuplet 5/4 {
      c'16 c'16 c'16 c'16 c'16
  }
}
}
\layout {
  \context {
    \Score
    proportionalNotationDuration = #(ly:make-moment 1/8)
  }
}
}

\score {
  \new RhythmicStaff {
    c'2
c'16 c'16 c'16 c'16
  \tuplet 5/4 {
      c'16 c'16 c'16 c'16 c'16
  }
}
}
\layout {
  \context {
    \Score
    proportionalNotationDuration = #(ly:make-moment 1/16)
  }
}
}

\score {
  \new RhythmicStaff {
    c'2
c'16 c'16 c'16 c'16
  \tuplet 5/4 {
      c'16 c'16 c'16 c'16 c'16
  }
}
}
Man muss beachten, dass die Referenzdauer nicht zu groß ist (wie die Achtel in dem Beispiel oben), denn dadurch werden die Noten so dicht gesetzt, dass sich eventuell sogar Notenköpfe von sehr kleinen Notenwerten überschneiden können. Die proportionale Notation nimmt üblicherweise mehr Platz ein als die klassische Platzverteilung. Der rhythmischen Klarheit muss ein eng gesetztes Notenbild geopfert werden.

In Folgenden soll betrachtet werden, wie sich überlappende rhythmische Aufteilungen am besten positioniert werden. Als Referenz wird das erste Beispiel herangezogen, zu welchem ein zweites System mit anderen rhythmischen Werten hinzugefügt wird:

\score {
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \tuplet 5/4 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
  \new RhythmicStaff {
    \tuplet 9/8 {
      c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
    }
  }
}
Die Platzaufteilung ist schlecht, weil die gleichlangen Noten des untersten Systems nicht gleichmäßig verteilt sind. Im klassischen Notensatz kommen komplexe rhythmische Verhältnisse wie dieses sehr selten vor, sodass der Notensatz nicht in Hinsicht auf sie optimiert ist. \texttt{proportionalNotationDuration} hilft in dieser Situation deutlich:

\begin{verbatim}
\score {
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \tuplet 5/4 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
  \new RhythmicStaff {
    \tuplet 9/8 {
      c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
    }
  }
}
\layout {
  \context {
    \Score
    proportionalNotationDuration = #(ly:make-moment 1/20)
  }
}
\end{verbatim}

Aber bei sehr genauer Betrachtung sind die Noten der zweiten Hälfte der Nonole doch immer noch eine Spur weiter gesetzt als die Noten der ersten Hälfte. Um wirklich gleichmäßige Abstände zu erzwingen, sollte auch noch die gleichmäßige Dehnung (\texttt{uniform-stretching}) angeschaltet werden, die eine Eigenschaft von \texttt{SpacingSpanner} ist:

\begin{verbatim}
\score {
  \new RhythmicStaff {
    c'2
    c'16 c'16 c'16 c'16
    \tuplet 5/4 {
      c'16 c'16 c'16 c'16 c'16
    }
  }
  \new RhythmicStaff {
    \tuplet 9/8 {
      c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
    }
  }
}
\end{verbatim}
Das Beispiel mit den zwei Systemen ist nun exakt nach den rhythmischen Werten der Noten gesetzt, sodass ein Zeitstrahl oder ähnliches eingefügt werden könnte.

Alle Einstellungen zur proportionalen Notation erwarten, dass die `uniform-stretching`-Eigenschaft des `SpacingSpanner`-Objekts auf wahr (`#t`) gesetzt wird. Andernfalls kann es vorkommen, dass bestimmte Abstände (etwa von unsichtbaren Noten) nicht richtig gesetzt werden.

Das `SpacingSpanner`-Objekt ist ein abstraktes Grob, dass sich im `Score`-Kontext befindet. Genauso wie die Einstellungen von `proportionalNotationDuration` können auch diese Veränderungen an den drei Stellen in der Quelldatei vorkommen: in der `\with`-Umgebung innerhalb von `Score`, in einer `\context`-Umgebung oder direkt im Notentext.

Standardmäßig gibt es nur ein `SpacingSpanner` pro `Score`. Das heißt, dass `uniform-stretching` für die gesamte Partitur (d.h. für die Reichweite von `Score`) entweder an- oder ausgeschaltet ist. Man kann allerdings in einer Partitur unterschiedliche Abschnitte mit verschiedenen Platzierungsverhalten definieren. Hierzu ist der Befehl `\newSpacingSection` da. Siehe auch Abschnitt 4.5.2 [Eine neuer Bereich mit anderen Abständen], Seite 538.

Im Folgenden soll gezeigt werden, wie sich der `Separating_line_group_engraver` auswirkt und warum er normalerweise für proportionale Notation ausgeschaltet wird. In diesem Beispiel wird verdeutlicht, dass vor jeder ersten Note eines Notensystems immer etwas zusätzlicher Platz gesetzt wird:

```plaintext
\paper {
  indent = #0
}

\new Staff {
  c'1
  \break
  c'1
}
```
Der gleiche horizontale zusätzliche Platz wird vor eine Noten gesetzt, wenn sie einer Taktart, einem Schlüssel oder einer Tonartbezeichnung folgt. Dieser Platz wird durch `Separating_line_group engraver` eingefügt; wenn wir ihn aus der Partitur entfernen, entfällt auch dieser zusätzliche Platz:

\begin{verbatim}
\paper {
    indent = #0
}
\new Staff \with {
    \remove "Separating_line_group_engraver"
} {
    c'1
    \break
    c'1
}
\end{verbatim}

Nichtmusikalische Elemente wie Takt- und Tonartangaben, Schlüssel und Versetzungszeichen sind problematisch in proportionaler Notation. Keine dieser Elemente hat eine rhythmische Dauer, aber alle brauchen horizontalen Platz. Das Problem wird auf unterschiedliche Weise gelöst.

Es ist manchmal möglich, Probleme mit Tonarten zu lösen, indem keine benutzt werden. Das ist durchaus eine ernstzunehmende Option, weil die meisten Partituren mit proportionaler Notation für heutige Musik geschrieben werden. Ähnliches gilt für Taktarten, insbesondere, wenn ein Zeitstrahl in die Partitur eingearbeitet werden soll. In den meisten Partituren kommt jedoch irgendeine Taktart vor. Schlüssel und Versetzungszeichen sind noch wichtiger; auf sie kann selten verzichtet werden.

Eine Lösungsmöglichkeit ist es, die `strict-note-spacing`-Eigenschaft des `SpacingSpanner`-Objekts zu benutzen. Zum Vergleich die beiden Partituren unten:

\begin{verbatim}
\new Staff {
    \set Score.proportionalNotationDuration = #(ly:make-moment 1/16)
    c''8
    c''8
    c''8
    \clef alto
    d'8
    d'2
}
\new Staff {
    \set Score.proportionalNotationDuration = #(ly:make-moment 1/16)
    \override Score.SpacingSpanner.strict-note-spacing = ##t
\end{verbatim}
Bei beiden handelt es sich um proportionale Notation, aber die Platzverteilung im oberen Beispiel ist zu weit wegen des Schlüsselwechsels. Die Platzverteilung des zweiten Beispiels dagegen bleibt rhythmisch korrekt. strict-note-spacing bewirkt, dass Takt- und Tonartbezeichnungen, Schlüssel und Versetzungssymbole keine Rolle bei der Berechnung der Abstände spielen.

Zusätzlich zu den hier vorgestellten Einstellungen gibt es noch eine Reihe von Möglichkeiten, die oft in proportionaler Notation benutzt werden. Dazu gehören:

- \override SpacingSpanner.strict-grace-spacing = ##t
- tupletFullLength = ##t
- \override Beam.breakable = ##t
- \override Glissando.breakable = ##t
- \override TextSpanner.breakable = ##t
- \remove "Forbid_line_break_engraver" in the Voice context

Diese Einstellungen bewirken, dass auch Verzierungsnoten proportional gesetzt werden, dass Klamern von rhythmischen Gruppen bis zu den Anfangs- und Endpunkten ausgedehnt werden und lassen dehnbare Objekte wie Balken und Glissandi auch über Taktstriche hinweg zu.

Siehe auch

Notationsreferenz: Abschnitt 4.5.2 [Eine neuer Bereich mit anderen Abständen], Seite 538.
Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.6 Die Musik auf weniger Seiten zwingen

Manchmal kommt es vor, dass nur ein oder zwei Systeme auf die nächste Seite geraten, obwohl es so aussieht, als ob auf der vorigen Seite genügend Platz ist, um diese Systeme auch noch unterzubringen.

4.6.1 Abstände anzeigen lassen

Die Dimensionen von vertikalen und horizontalen Platzierungsvariablen, die veränderbar sind, lassen sich mit ihren aktuellen Werten im Notentext anzeigen, wenn man die Funktion `annotate-spacing` in der `paper`-Umgebung einschaltet:

```lilypond
#(set-default-paper-size "a6" 'landscape)
\book {
  \score { { c4 } }
  \paper { annotate-spacing = ##t }
}
```


1 Punkt \( = (25.4/72.27) \) mm

1 Notenlinienzwischenraum \( = (\text{staff-size})/4 \) pts
\( = (\text{staff-size})/4 \times (25.4/72.27) \) mm

In diesem Fall ist ein `staff-space` (Notenlinienzwischenraum) etwa gleich 1.757 mm. Deshalb entspricht der Wert von 95.75 `staff-space` für `paper-height` (Papierhöhe) 105 mm, die Höhe eines quer gelegten A6-Papiers. Die Paare \((a,b)\) sind Intervalle, wobei \(a\) der untere Rand und \(b\) der obere Rand des Intervalls.
Siehe auch

Notationsreferenz: Abschnitt 4.2.2 [Die Notensystemgröße einstellen], Seite 511,
Schnipsel: Abschnitt “Spacing” in Schnipsel.

4.6.2 Abstände verändern


Neben Rändern gibt es einige weitere Optionen, Platz zu sparen:

• LilyPond kann die Systeme so dicht wie möglich platziern (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 {
   system-system-spacing = #'((padding . 0) (basic-distance . 0.1))
   ragged-last-bottom = ##f
   ragged-bottom = ##f
}

• Die Anzahl der Systeme kann erzwungen werden. Das kann auf zwei Arten helfen: wenn einfach nur ein Wert gesetzt wird, auch wenn es die gleiche Anzahl ist, die auch schon vorher von LilyPond erstellt wurde, kann manchmal dazu führen, dass mehr Systeme auf eine Seite gesetzt werden. Das liegt daran, dass ein Schritt im Notensatz ausgelassen wird, der die Seitenverteilung nur grob einschätzt, sodass eine bessere Seitenverteilung entsteht. Auch wenn man eine Verringerung der Anzahl an Systemen erzwingt, kann oft eine Seite eingespart werden. Wenn LilyPond die Musik etwa auf 11 Systeme verteilt, kann man die Benutzung von nur 10 Systemen erzwingen.

\paper {
   system-count = #10
}

• Die Anzahl der Seiten kann erzwungen werden. Beispielsweise erzwingt folgender Code ein Layout mit zwei Seiten:

\paper {
   page-count = #2
}


Ein anderes Beispiel ist es, Dynamik-Zeichen, die besonders weit „hervorstehen“, zu verschieben.

\e4 \c \g \f \c
\e4 \c \g \-\tweak X-offset #-2.7 -\tweak Y-offset #2.5 \f \c
• Die horizontalen Abstände können mit der SpacingSpanner-Eigenschaft verändert werden. Siehe Abschnitt 4.5.3 [Horizontale Abstände verändern], Seite 538, für Einzelheiten. Dieses Beispiel zeigt die normalen Abstände:

\begin{verbatim}
\score {
    \relative {
        g'4 e e2 |
        f4 d d2 |
        c4 d e f |
        g4 g g2 |
        g4 e e2 |
    }
}
\end{verbatim}

Das nächste Beispiel verändert common-shortest-duration (die häufigste kürzeste Note) von 1/4 zu 1/2. Die Viertelnote ist dennoch die häufigste Note in diesem Abschnitt, sodass der Notentext zusammengedrängt, wird, wenn eine Halbe als Standard angegeben wird:

\begin{verbatim}
\score {
    \relative {
        g'4 e e2 |
        f4 d d2 |
        c4 d e f |
        g4 g g2 |
        g4 e e2 |
    }
    \layout {
        \context {
            \Score
            \override SpacingSpanner.common-shortest-duration = #(ly:make-moment 1/2)
        }
    }
}
\end{verbatim}

Die common-shortest-duration-Eigenschaft kann nicht dynamisch verändert werden, darum muss sie immer in der \context-Umgebung definiert werden und wirkt sich somit auf eine ganze \score-Umgebung aus.

Siehe auch

Notationsreferenz: Abschnitt 4.1 [Seitenlayout], Seite 498, Abschnitt 4.5.3 [Horizontale Abstände verändern], Seite 538.
Schnipsel: Abschnitt “Spacing” in Schnipsel.
5 Standardeinstellungen verändern


Intern benutzt LilyPond Scheme (ein LISP-Dialekt), um eine Infrastruktur zur Verfügung zu stellen. Wenn Layoutentscheidungen verändert werden sollen, müssen auf die programminternen Prozesse zugegriffen werden, worzu Scheme-Code benötigt wird. Scheme-Abschnitte werden in einer LilyPond-Quelldatei mit einer Raute # begonnen.¹

5.1 Interpretationskontexte

Dieser Abschnitt erklärt, was Kontexte sind und wie man sie verändern kann.

Siehe auch

Handbuch zum Lernen: Abschnitt “Kontexte und Engraver” in Handbuch zum Lernen.
Schnipsel: Abschnitt “Contexts and engravers” in Schnipsel.

5.1.1 Was sind Kontexte?

Kontexte sind hierarchisch geordnet:

Score – der Vater aller Kontexte

Score (Partitur) ist der höchste Notationskontext. Kein anderer Kontext kann einen Score-Kontext enthalten. Im Normalfall kümmert sich der Score-Kontext um die Verwaltung der Taktarten und sorgt dafür, dass Elemente wie Schlüssel und Taktart- oder Tonartbezeichnungen über die Systeme hinweg aneinander ausgerichtet sind.

Ein Score-Kontext wird eingerichtet, wenn eine \score {...} oder \layout {...}-Umgebung interpretiert wird.

Oberste Kontexte – Container für Systeme

Diese Kontexte fassen Systeme zu Gruppen zusammen und werden darum hier als Systemgruppen bezeichnet (engl. staffgroup).

StaffGroup

Gruppiert Systeme und fügt eine eckige Klammer auf der linken Seite hinzu. Die Taktstriche der enthaltenen Systeme werden vertikal miteinander verbunden. StaffGroup besteht nur aus

¹ Abschnitt “Scheme-Tutorium” in Extending enthält eine kurze Übung, wie man Zahlen, Listen, Zeichenketten und Symbole in Scheme notiert.
einer Ansammlung von Systemen mit einer eckigen Klammer zu Beginn der Zeile und durchgezogenen Taktstriche.

ChoirStaff
Entspricht StaffGroup, außer dass die Taktstriche der enthaltenen Systeme nicht vertikal miteinander verbunden sind.

GrandStaff

PianoStaff
Entspricht GrandStaff, hat aber zusätzlich Unterstützung für Instrumentenbezeichnungen zu Beginn jeder Systemgruppe.

**Mittlere Kontexte – Systeme**
Diese Kontexte stellen verschiedene Arten einzelner Notationssysteme (engl. staff) dar.

Staff
Kümmert sich um Schlüssel, Taktstriche, Tonarten und Versetzungszeichen. Er kann Voice-Kontexte enthalten.

RhythmicStaff
Entspricht Staff, aber dient zur Notation von Rhythmen: Tonhöhen werden ignoriert und die Noten auf einer einzigen Linie ausgegeben.

TabStaff
Ein Kontext um Tabulaturen zu erstellen. Die Standardeinstellung ist eine Gitarrentabulatur mit sechs Notenlinien.

DrumStaff

VaticanaStaff
Entspricht Staff, aber eignet sich besonders zum Notensatz des Gregorianischen Chorals.

MensuralStaff
Entspricht Staff, aber eignet sich zum Notensatz von Noten in der Mensuralnotation.

**Unterste Kontexte – Stimmen**

Voice

VaticanaVoice
Entspricht Voice, aber eignet sich besonders zum Notensatz des Gregorianischen Chorals.

MensuralVoice
Entspricht Voice, aber mit Änderungen, um Mensuralnotation setzen zu können.

Lyrics

DrumVoice
Der Stimmenkontext in einem Perkussionssystem.

**FiguredBass**
Der Kontext, in dem Generalbassziffern (BassFigure-Objekte) gesetzt werden, die in der \figuremode-Umgebung notiert werden.

**TabVoice**
Dieser Stimmenkontext wird in einer Tabulatur (TabStaff-Kontext) benutzt. Er wird normalerweise implizit erstellt.

**CueVoice**

**ChordNames**
Ausgabe von Akkordsymbolen.

### 5.1.2 Kontexte erstellen und referenzieren

In Partituren mit einer Stimme und einem System werden die Kontexte normalerweise automatisch erstellt. In kompliziereren Partituren muss man sie aber direkt erstellen. Es gibt drei Möglichkeiten, Kontexte zu erstellen:

- Der einfachste Befehl ist \new. Er wird zusammen mit dem Kontextnamen vor einem musikalischen Ausdruck eingesetzt, etwa
  
  \new Kontext musik. Ausdruck

  wobei Kontext eine Kontextbezeichnung (wie Staff oder Voice) ist. Dieser Befehl erstellt einen neuen Kontext und beginnt mit der Auswertung von musik. Ausdruck innerhalb dieses Kontextes.

  Eine praktische Anwendung von \new ist eine Partitur mit vielen Systemen. Jede Stimme wird auf einem eigenen System notiert, das mit \new Staff begonnen wird.

  `<<
   \new Staff \{ c4 c \}
   \new Staff \{ d4 d \}
  >>`

  Der \new-Befehl kann den Kontext auch benennen:

  \new Kontext = ID musik. Ausdruck

  Dieser vom Benutzer definierte Name wird aber auch nur wirklich benutzt, wenn nicht vorher schon der gleiche Name definiert worden ist.

- Ähnlich dem \new-Befehl wird auch mit dem \context-Befehl ein musikalischer Ausdruck in einen Kontext umgeleitet. Diesem Kontext wird ein expliziter Name zugewiesen. Die Syntax lautet:

  \context Kontext = ID musik. Ausdruck

  Diese Art von Befehl sucht nach einem existierenden Kontext vom Typus Kontext mit der Bezeichnung ID. Wenn ein derartiger Kontext nicht existiert, wird ein neuer Kontext mit
der entsprechenden Bezeichnung erstellt. Das ist nützlich, wenn auf den Kontext später zurückverwiesen werden soll. Um etwa Gesangstext zu einer Melodie hinzuzufügen, wird die Melodie in einem bezeichneten Kontext notiert:

\context Voice = "Tenor" musik. Ausdruck

sodass der Text an den Noten ausgerichtet werden kann:

\new Lyrics \lyricsto "Tenor" Gesangstext

Eine andere Möglichkeit für bezeichnete Kontexte ist es, zwei unterschiedliche musikalische Ausdrücke in einen Kontext zu verschmelzen. Im nächsten Beispiel werden Artikulationszeichen und Noten getrennt notiert:

\new Staff \context Voice = "A" \Noten\context Voice = "A" \Artik

Dann werden sie kombiniert, indem sie dem selben Voice-Kontext zugewiesen werden:

<< \new Staff \context Voice = "A" \Noten \context Voice = "A" \Artik >>

Durch diesen Mechanismus ist es möglich eine Urtextausgabe zu erstellen, mit der optionalen Möglichkeit, bestimmte zusätzliche Artikulationszeichen zu den gleichen Noten hinzuzufügen und so eine editierte Ausgabe zu erhalten.

• Der dritte Befehl, um Kontexte zu erstellen, ist:

\context Kontext musik. Ausdruck

Dies entspricht dem \context mit = ID, aber hier wird ein beliebiger Kontext des Typs Kontext gesucht und der musikalische Ausdruck darin ausgewertet, unabhängig von der Bezeichnung, die dem Kontext gegeben wurde.

Diese Variante wird bei musikalischen Ausdrücken benutzt, die auf verschiedenen Ebenen interpretiert werden können. Beispielsweise der \applyOutput-Befehl (siehe Abschnitt "Eine Funktion auf alle Layout-Objekte anwenden" in Extending). Ohne einen expliziten \context wird die Ausgabe normalerweise einem Voice-Kontext zugewiesen:

\applyOutput #'Kontext #Funktion % auf Voice anwenden

Damit aber die Funktion auf Score- oder Staff-Ebene interpretiert wird, muss folgende Form benutzt werden:

\applyOutput Score #Funktion
\applyOutput Staff #Funktion

5.1.3 Kontexte am Leben halten

Kontexte werden normalerweise am ersten musikalischen Moment beendet, an dem sie nichts mehr zu tun haben. Ein Voice-Kontext stirbt also sofort, wenn keine Ereignisse mehr auftreten, Staff-Kontexte sobald alle in ihnen enthaltenen Voice-Kontexte keine Ereignisse mehr aufweisen usw. Das kann Schwierigkeiten ergeben, wenn auf frühere Kontexte verwiesen werden soll, die in der Zwischenzeit schon gestorben sind, beispielsweise wenn man Systemwechsel mit \change-Befehlen vornimmt, wenn Gesangstext einer Stimme mit dem \lyricsto-Befehl zugewiesen wird oder wenn weitere musikalische Ereignisse zu einem früheren Kontext hinzugefügt werden sollen.
Es gibt eine Ausnahme dieser Regel: genau ein Voice-Kontext innerhalb eines Staff-Kontextes oder einer <<...>>-Konstruktion bleibt immer erhalten bis zum Ende des Staff-Kontextes oder der <<...>>-Konstruktion, der ihn einschließt, auch wenn es Abschnitte gibt, in der er nichts zu tun hat. Der Kontext, der erhalten bleibt ist immer der erste, der in der ersten enthaltenden {...}-Konstruktion angetroffen wird, wobei <<...>>-Konstruktionen ignoriert werden.

Jeder Kontext kann am Leben gehalten werden, indem man sicherstellt dass er zu jedem musikalischen Moment etwas zu tun hat. Staff-Kontexte werden am Leben gehalten, indem man sicherstellt, dass eine der enthaltenen Stimmen am Leben bleibt. Eine Möglichkeit, das zu erreichen, ist es, unsichtbare Pause zu jeder Stimme hinzufügen, die am Leben gehalten werden soll. Wenn mehrere Stimmen sporadisch benutzt werden sollen, ist es am sichersten, sie alle am Leben zu halten und sich nicht auf die Ausnahmeregel zu verlassen, die im vorigen Abschnitt dargestellt wurde.

Im folgenden Beispiel werden sowohl Stimme A als auch B auf diese Weise für die gesamte Dauer des Stückes am Leben gehalten.

```music
musicA = \relative { d''4 d d d }
musicB = \relative { g'4 g g g }
keepVoicesAlive = {
  <<
    \new Voice = "A" { s1*5 } % Keep Voice "A" alive for 5 bars
    \new Voice = "B" { s1*5 } % Keep Voice "B" alive for 5 bars
  >>
}
music = {
  \context Voice = "A" {
    \voiceOneStyle
    \musicA
  }
  \context Voice = "B" {
    \voiceTwoStyle
    \musicB
  }
  \context Voice = "A" { \musicA }
  \context Voice = "B" { \musicB }
  \context Voice = "A" { \musicA }
}
\score {
  \new Staff <<
    \keepVoicesAlive
    \music
  >>
}
```

Das nächste Beispiel zeigt eine Melodie, die zeitweise unterbrochen wird und wie man den entsprechenden Gesangstext mit ihr verknüpfen kann, indem man die Stimme am Leben hält.
In wirklichen Situationen würden Begleitung und Melodie natürlich aus mehreren Abschnitten bestehen.

```latex
melody = \relative { a'4 a a a }
accompaniment = \relative { d'4 d d d }
words = \lyricmode { These words follow the melody }
\score {
  \new Staff = "music" {
    \new Voice = "melody" {
      \voiceOne
      s1*4 % Keep Voice "melody" alive for 4 bars
    }
    \new Voice = "accompaniment" {
      \voiceTwo
      \accompaniment
    }
    \context Voice = "melody" { \melody }
    \context Voice = "accompaniment" { \accompaniment }
  }
  \context Voice = "accompaniment" { \accompaniment }
  \context Voice = "melody" { \melody }
  \context Voice = "accompaniment" { \accompaniment }
}
\new Lyrics \with { alignAboveContext = "music" }
\lyricsto "melody" { \words }
```

Eine Alternative, die in manchen Umständen besser geeignet sein kann, ist es, einfach unsichtbare Pausen einzufügen, um die Melodie mit der Begleitung passend auszurichten:

```latex
melody = \relative {
  s1 % skip a bar
  a'4 a a a
  s1 % skip a bar
  a4 a a a
}
accompaniment = \relative {
  d'4 d d d
  d4 d d d
```

These words follow the melody
Kapitel 5: Standardeinstellungen verändern

5.1.4 Umgebungs-Plugins verändern

Notationskontexte (wie Score oder Staff) speichern nicht nur Eigenschaften, sie enthalten auch Plugins („engraver“ genannt), die die einzelnen Notationselemente erstellen. Ein Voice-Kontext enthält beispielsweise einen Note_heads_engraver, der die Notenköpfe erstellt, und ein Staff-Kontext einen Key_engraver, der die Vorzeichen erstellt.

Eine vollständige Erklärung jedes Plugins findet sich in Referenz der Interna: ➔ Translation ➔ Engravers. Alle Kontexte sind erklärt in Referenz der Interna: ➔ Translation ➔ Context, wobei die in diesem Kontext vorkommenden Engraver aufgelistet sind.

Es kann teilweise nötig sein, diese Engraver umzupositionieren. Das geschieht, indem man einen neuen Kontext mit \new oder \context beginnt und ihn dann verändert:

```
\new context \with {
\consists ...
\consists ...
\remove ...
\remove ...
etc.
}
{
..Noten..
}
```

... steht hier für die Bezeichnung des Engravers. \consists fügt einen Engraver hinzu und \remove entfernt ihn. Es folgt ein einfaches Beispiel, in dem der Time_signature_engraver

```
d4 d d d
d4 d d d
}
words = \lyricmode { These words follow the melody }
\score {
<<
\new Staff = "music" {
<<
\new Voice = "melody" {
\voiceOne
\melody
}
\new Voice = "accompanyment" {
\voiceTwo
\accompaniment
}
>>
}
\new Lyrics \with { alignAboveContext = "music" }
\lyricsto "melody" { \words }
>>
```

These words follow the melody

\[\begin{array}{c}
\text{These words follow the mel-o-dy}
\end{array}\]
Kapitel 5: Standardeinstellungen verändern

(Engraver für den Takt) und der Clef_engraver (Engraver für den Schlüssel) aus dem Staff-Kontext entfernt werden:

```
<<
\new Staff \relative {
 f'2 g
}
\new Staff \with {
 \remove "Time_signature_engraver"
 \remove "Clef_engraver"
} \relative {
 f'2 g2
}
>>
```

Das zweite Notensystem enthält keine Taktangabe und keinen Notenschlüssel. Das ist eine recht brutale Methode, Objekte zu verstecken, weil es sich auf das gesamte System auswirkt. Diese Methode beeinflusst auch die Platzauflösung, was erwünscht sein kann. Vielfältigere Methoden, mit denen Objekte unsichtbar gemacht werden können, finden sich in Abschnitt “Sichtbarkeit und Farbe von Objekten” in Handbuch zum Lernen.

Das nächste Beispiel zeigt eine Anwendung in der Praxis. Taktstriche und Taktart werden normalerweise in einer Partitur synchronisiert. Das geschieht durch Timing_translator und Default_bar_line_engraver. Diese Plugins sorgen sich um die Verwaltung der Taktzeiten und die Stelle innerhalb des Taktes, zu dem eine Note erscheint usw. Indem man diese Engraver aus dem Score-Kontext in den Staff-Kontext verschiebt, kann eine Partitur erstellt werden, in welcher jedes System eine unterschiedliche Taktart hat:

```
\score {
<<
 \new Staff \with {
 \consists "Timing_translator"
 \consists "Default_bar_line_engraver"
 } {
 \time 3/4
 c4 c c c c c
 }
 \new Staff \with {
 \consists "Timing_translator"
 \consists "Default_bar_line_engraver"
 } {
 \time 2/4
 c4 c c c c c
 }
>>
 \layout {
 \context { 
 \Score
```
Bekannte Probleme und Warnungen

Die Reihenfolge, in der die Engraver definiert werden, ist die Reihenfolge, in welcher sie aufgerufen werden, um ihre Verarbeitung vorzunehmen. Normalerweise spielt die Reihenfolge, in welcher die Engraver angegeben werden, keine Rolle, aber in einigen Spezialfällen ist die Rei- henfolge sehr wichtig. Das kann beispielsweise vorkommen, wenn ein Engraver eine Eigenschaft erstellt und ein anderer von ihr liest, oder ein Engraver erstellt ein Grob und ein anderer wertet es aus.

Folgende Reihenfolgen müssen beachtet werden:
- der Bar_engraver muss normalerweise zuerst kommen,
- der New_fingering_engraver muss vor dem Script_column_engraver kommen,
- der Timing_translator muss vor dem Bar_number_engraver kommen.

Siehe auch


5.1.5 Die Standardeinstellungen von Kontexten ändern

Kontext- und Grob-Eigenschaften können mit den Befehlen \set und \override verändert werden, wie beschrieben in Abschnitt 5.3 [Eigenschaften verändern], Seite 572. Diese Befehle erstellen musikalische Ereignisse, damit die Veränderungen zum Zeitpunkt der Verarbeitung in den Noten erscheinen.


Alle Kontexte des gleichen Typs verändern

Die Kontexteinstellungen, die standardmäßig in Score, Staff, Voice und anderen Kontexten eingesetzt werden, können in einer \context-Umgebung innerhalb einer beliebigen \layout-Umgebung spezifiziert werden. Die \layout-Umgebung sollte innerhalb der \score (Partitur) stehen, auf die sie sich bezieht, nach den Noten.

\layout {
    \context {
        \Voice
            [Kontexteinstellungen für alle Voice-Kontexte]
Kapitel 5: Standardeinstellungen verändern

Folgende Einstellungstypen können angegeben werden:

- Ein \override-Befehl, aber ohne die Kontextbezeichnung:
  \score {
  \relative {
    a'4"Thicker stems" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Staff
      \override Stem.thickness = #4.0
    }
  }
}

\[Thicker stems\]

- Eine Kontexteigenschaft kann direkt gesetzt werden:
  \score {
  \relative {
    a'4"Smaller font" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Staff
      fontSize = #-4
    }
  }
}

\[Smaller font\]

- Ein vordefinierter Befehl wie etwa \dynamicUp oder ein musikalischer Ausdruck wie \accidentalStyle dodecaphonic:
  \score {
    \relative {
      a'4"Dynamics above" a a a
      a4 a a\ff a
    }
  }

\[Dynamics above\]
Kapitel 5: Standardeinstellungen verändern

Eine vom Benutzer definierte Variable, die eine \with-Umgebung enthält; zu Details der \with-Umgebung, siehe [Nur einen bestimmten Kontext verändern], Seite 563.

StaffDefaults = \with {
  fontSize = #-4
}

\score {
  \new Staff {
    \relative {
      a'4"Smaller font" a a a
      a4 a a a
    }
  }
  \layout {
    \context {
      \Staff
      \StaffDefaults
    }
  }
}


\score {
  \new Staff {
    \relative {
      \Staff
    }
  }
}
Nur einen bestimmten Kontext verändern

Die Kontexteigenschaften nur eines bestimmten Kontextes können mit einer \with-Umgebung geändert werden. Alle anderen Vorkommen des gleichen Kontexts behalten ihre Standardeinstellungen, möglicherweise durch Einstellungen in \layout-Umgebungen verändert. Die \with-Umgebung muss direkt nach dem Befehl \new Kontext-Typ gesetzt werden:

\new Staff
\with {
[enthält Einstellungen nur für diesen spezifischen Kontext]
} {
...
}

Folgende Arten von Einstellungen können angegeben werden:

• Ein \override-Befehl, dessen Kontextbezeichnung ausgelassen wird:

\score {
\new Staff {
\new Voice
\with {
\override Stem.thickness = #4.0
}
{
\relative {
 a'4"Thick stems" a a a
 a4 a a a
}
}
}
}

• Eine Kontexeigenschaft direkt einstellen:

\score {
<<
Ein vordefinierter Befehl wie etwa \dynamicUp

\score {
   \new Staff {
      \relative {
         a'4\"Dynamics below" a a a
         a4 a a a
      }
   }
}
\new Staff
\with { \accidentalStyle dodecaphonic }
{n}
\new Voice
\with { \dynamicUp }
{n}
\relative {
   a'4\"Dynamics above" a a a
   a4 a a\ff a
}
}
Rangfolge von Kontextwerten
Der Wert einer Eigenschaft, die zu einer bestimmten Zeit aktiv ist, wird wie folgend bestimmt:

- wenn ein \override- oder \set-Befehl in der Eingabe aktiv ist, wird dieser Wert benutzt,
- ansonsten wird der Standardwert aus einer \with-Umgebung zu Beginn des Kontextes benutzt,
- ansonsten wird der Standardwert aus der letzten passenden \context-Umgebung in der letzten \layout-Umgebung benutzt,
- ansonsten wird der Standardwert von LilyPond eingesetzt.

Siehe auch
Handbuch zum Lernen: Abschnitt “Kontexteigenschaften verändern” in Handbuch zum Lernen.
Notationsreferenz: Abschnitt 5.1.1 [Was sind Kontexte?], Seite 552, [Unterste Kontexte – Stimmen], Seite 553, Abschnitt 5.3.2 [Der set-Befehl], Seite 572, Abschnitt 5.3.3 [Der override-Befehl], Seite 574, Abschnitt 4.2.1 [Die layout-Umgebung], Seite 509.

5.1.6 Neue Kontexte definieren
Bestimme Kontexte, wie Staff oder Voice, werden erstellt, indem man sie mit einer Musikumgebung aufruft. Es ist aber auch möglich, eigene neue Kontexte zu definieren, in denen dann unterschiedliche Engraver benutzt werden.


Diese Einstellungen werden innerhalb der \context-Umgebung innerhalb der \layout-Umgebung definiert:

\layout {
  \context {
    ...
  }
}

Der Beispielcode des folgenden Abschnitttest muss anstelle der Punkte im vorigen Beispiel eingesetzt werden.
Zuerst ist es nötig eine Bezeichnung für den neuen Kontext zu definieren:

\name ImproVoice

Weil dieser neue Kontext ähnlich wie Voice ist, sollen die Befehle, die in Voice-Kontexten funktionieren, auch in dem neuen Kontext funktionieren. Das wird erreicht, indem der Kontext als Alias Voice erhält:

\alias Voice
Der Kontext gibt Noten und Text aus, darum müssen wir die Engraver hinzufügen, die für diese Aktionen zuständig sind:

\consists "Note_heads_engraver"
\consists "Text_engraver"

aber die Noten sollen nur auf der mittleren Linie ausgegeben werden:

\consists "Pitch_squash_engraver"
squashedPosition = #0

Der Pitch_squash_engraver verändert Notenköpfe (die vom Note_heads_engraver erstellt werden) und setzt ihre vertikale Position auf den Wert von squashedPosition, in diesem Fall ist das die Mittellinie.

Die Noten sehen wie ein Querstrich aus und haben keine Hälse:

\override NoteHead.style = #'slash
\hide Stem

Alle diese Engraver müssen zusammenarbeiten, und das wird erreicht mit einem zusätzlichen Plugin, das mit dem Befehl \type gekennzeichnet werden muss. Dieser Typ solle immer Engraver_group lautet:

\type "Engraver_group"

Alles zusammen haben wir folgende Einstellungen:

\context {
\name ImproVoice
\type "Engraver_group"
\consists "Note_heads_engraver"
\consists "Text_engraver"
\consists "Pitch_squash_engraver"
squashedPosition = #0
\override NoteHead.style = #'slash
\hide Stem
\alias Voice
}

Kontexte sind hierarchisch. Wie wollen, dass ImproVoice sich als Unterkontext von Staff erkennt, wie eine normale Stimme. Darum wird die Definition von Staff mit dem \accepts-Befehl verändert:

\context {
\Staff
\accepts ImproVoice
}

Das Gegenteil von \accepts ist \denies (verbietet), was manchmal gebraucht werden kann, wenn schon existierende Kontext-Definitionen wieder benutzt werden sollen.

Beide Definitionen müssen in die \layout-Umgebung geschrieben werden:

\layout {
\context {
\name ImproVoice
...}
\context {
\Staff
\accepts "ImproVoice"
}
Jetzt kann die Notation zu Beginn des Abschnitts folgendermaßen notiert werden:

\relative {
  a'4 d8 bes8
  \new ImproVoice {
    c4"ad lib" c
    c4 c"ausziehen"
    c c_"während des Spielens :)"
  }
  a1
}

5.1.7 Reihenfolge des Kontextlayouts


Die „accepts“-Liste eines Kontextes kann mit dem Befehlen \accepts und \denies verändert werden. \accepts fügt einen Kontext zur „accepts“-Liste, und \denies entfernt einen Kontext aus der Liste. Akkordbezeichnungen sollen beispielsweise normalerweise nicht innerhalb eines Staff-Kontextes geschachtelt werden, sodass der ChordNames-Kontext nicht automatisch in der „accepts“-Liste des Staff-Kontextes geführt ist. Wenn er aber benötigt wird, kann er hinzugefügt werden:

\score {
  \new Staff {
    c' d' e' f'
    \chords { d1:m7 b1:min7.5- }
  }
}

\score {
  \new Staff {
    c' d' e' f'
    \chords { d1:m7 b1:min7.5- }
  }
  \layout {
    \context {
      \Staff
      \accepts "ChordNames"
    }
  }
}
\deny wird vorrangig eingesetzt, wenn ein neuer Kontext basierend auf einem existierenden erstellt wird, aber sein Schachtelungsverhalten sich unterscheidet. Der VaticanaStaff-Kontext beispielsweise basiert auf dem Staff-Kontext, hat aber den VaticanaVoice-Kontext anstellt des Voice-Kontexts in seiner „accepts“-Liste.


Manchmal soll ein Kontext nur für einen kurzen Moment existieren, ein gutes Beispiel etwa ein System für ein Ossia. Das wird normalerweise erreicht, indem man die Kontextdefinition an der richtigen Stelle parallel mit dem existierenden Abschnitt der Hauptnoten anlegt. Standardmäßig wird der neue Kontext unter den existierenden Kontexten angelegt. Um ihn aber über dem Kontext mit der Bezeichnung „Hauptstimme“ zu positionieren, sollte er folgenderweise erstellt werden:

\new Staff \with { alignAboveContext = "Hauptstimme" }


Beispiele, die diese Neuordnung von temporären Kontexten zeigen, finden sich an anderen Stellen; siehe Abschnitt “Musikalische Ausdrücke ineinander verschachteln” in Handbuch zum Lernen, Abschnitt 1.6.2 [Einzelne Systeme verändern], Seite 181, und Abschnitt 2.1.2 [Techniken für die Gesangstextnotation], Seite 254.

Siehe auch

Handbuch zum Lernen: Abschnitt “Musikalische Ausdrücke ineinander verschachteln” in Handbuch zum Lernen.

Notationsreferenz: Abschnitt 1.6.2 [Einzelne Systeme verändern], Seite 181, Abschnitt 2.1.2 [Techniken für die Gesangstextnotation], Seite 254.


5.2 Die Referenz der Programminterna erklärt

5.2.1 Zurechtfinden in der Programmreferenz

Arbeit mit der Referenz der Interna soll hier an einigen Beispiel illustriert werden. Die Referenz der Interna existiert nur auf Englisch, darum sind auch die Beispiele dieses Abschnittes nicht übersetzt.

Folgende Aufgabe wird bearbeitet: Der Fingersatz aus dem Beispiel unten soll verändert werden:

\new Staff \with { alignAboveContext = "Hauptstimme" }

\stemUp

\f
In der Dokumentation über Fingersatz (in [Fingersatzanweisungen], Seite 205) gibt es folgenden Abschnitt:

Siehe auch:

Referenz der Interna: Abschnitt “Fingering” in Referenz der Interna.

Die Referenz der Interna gibt es als HTML-Dokument. Sie sollten sie als HTML-Dokument lesen, entweder online oder indem Sie die HTML-Dokumentation herunterladen. Dieser Abschnitt ist sehr viel schwieriger zu verstehen, wenn Sie die PDF-Version verwenden.

Gehen Sie über diesen Link zum Abschnitt Abschnitt “Fingering” in Referenz der Interna. Oben auf der Seite findet sich:

Fingering objects are created by: Abschnitt “Fingering_engraver” in Referenz der Interna and Abschnitt “New_fingering_engraver” in Referenz der Interna.

Indem Sie die Links in der Referenz der Interna folgen, können Sie verfolgen, wie LilyPond intern arbeitet:

• Abschnitt “Fingering” in Referenz der Interna: Abschnitt “Fingering” in Referenz der Interna objects are created by: Abschnitt “Fingering_engraver” in Referenz der Interna
• Abschnitt “Fingering_engraver” in Referenz der Interna: Music types accepted: Abschnitt “fingering-event” in Referenz der Interna
• Abschnitt “fingering-event” in Referenz der Interna: Music event type fingering-event is in Music expressions named Abschnitt “FingeringEvent” in Referenz der Interna

Fingersatz-Objekte werden also durch den Fingering_engraver erstellt, welcher folgende Musikereignistypen akzeptiert: fingering-event. Ein Musikereignis vom Typ fingering-event ist ein musikalischer Ausdruck mit der Bezeichnung Abschnitt “FingeringEvent” in Referenz der Interna.

Dieser Pfad geht genau die entgegengesetzte Richtung von LilyPonds Wirkungsweise: er beginnt bei der graphischen Ausgabe und arbeitet sich voran zur Eingabe. Man könnte auch mit einem Eingabe-Ereignis starten und dann die Links zurückverfolgen, bis man zum Ausgabe-Objekt gelangt.


5.2.2 Layout-Schnittstellen


Die Seite zu Fingering enthält Definitionen für das Fingering-Objekt. Auf der Seite steht etwa:

padding (dimension, in staff space):
0.5

was bedeutet, dass der Abstand zu anderen Objekten mindestens 0.5 Notenlinienabstände beträgt.
Jedes Layoutobjekt kann mehrere Funktionen sowohl als typographisches als auch als Notationselement einnehmen. Das Fingersatzobjekt beispielsweise hat folgende Aspekte:

- Seine Größe ist unabhängig von der horizontalen Platzaufteilung, anders als etwa bei Legatobögen.
- Es handelt sich um Text, normalerweise sehr kurz.
- Dieser Text wird durch ein Glyph einer Schriftart gesetzt, anders als bei Legatobögen.
- Der Mittelpunkt des Symbols sollte horizontal mit dem Mittelpunkt des Notenkopfes ausgerichtet werden.
- Vertikal wird das Objekt neben die Note und das Notensystem gesetzt.
- Die vertikale Position wird auch mit anderen Textelementen abgeglichen.

Jeder dieser Aspekte findet sich in sogenannten Schnittstellen (engl. interface), die auf der Abschnitt “Fingering” in Referenz der Interna-Seite unten aufgelistet sind:


Es wurde immer von einem Fingering-Objekt gesprochen, aber eigentlich handelt es sich nicht um sehr viel. Die Initialisierungsdatei scm/define-grobs.scm zeigt den Inhalt dieses „Objekts“ (zu Information, wo diese Dateien sich finden siehe Abschnitt „Mehr Information“ in Handbuch zum Lernen):

(Fingering
  . ((padding 0.5)
    (avoid-slur around)
    (slur-padding 0.2)
    (staff-padding 0.5)
    (self-alignment-X 0)
    (self-alignment-Y 0)
    (script-priority 100)
    (stencil . ,ly:text-interface::print)
    (direction . ,ly:script-interface::calc-direction)
    (font-encoding . fetaText)
    (font-size -5) ; don't overlap when next to heads.
    (meta . ((class Item)
      (interfaces . (finger-interface
        font-interface
        text-script-interface
        text-interface
        side-position-interface
        self-alignment-interface
        item-interface)))))

Wie man sehen kann, ist das Fingersatzobjekt nichts anderes als eine Ansammlung von Variablen, und die Internetseite der Referenz der Interna ist direkt aus diesen Anweisungen generiert.
5.2.3 Die Grob-Eigenschaften

Die Position der 2 aus dem Beispiel unten soll also geändert werden:

\begin{verbatim}
c-2 \stemUp f
\end{verbatim}

Weil die 2 vertikal an der zugehörigen Note ausgerichtet ist, müssen wir uns mit der Schnittstelle auseinander setzen, die diese Positionierung veranlasst. Das ist hier side-position-interface. Auf der Seite für diese Schnittstelle heißt es:

`side-position-interface`

Position a victim object (this one) next to other objects (the support). The property direction signifies where to put the victim object relative to the support (left or right, up or down?)

Darunter wird die Variable padding (Verschiebung) beschrieben:

`padding` (dimension, in staff space)

Add this much extra space between objects that are next to each other.

Indem man den Wert von padding erhöht, kann die Fingersatzanweisung weiter weg von der Note gesetzt werden. Dieser Befehl beispielsweise fügt drei Notenlinienzwischenräume zwischen die Zahl und den Notenkopf:

\begin{verbatim}
\once \override Voice.Fingering.padding = #3
\end{verbatim}

Wenn dieser Befehl in den Quelltext eingefügt wird, bevor der Fingersatz notiert ist, erhält man folgendes:

\begin{verbatim}
\once \override Voice.Fingering.padding = #3
c-2 \stemUp f
\end{verbatim}

In diesem Fall muss die Veränderung speziell für den Voice-Kontext definiert werden. Das kann auch aus der Referenz der Interna entnommen werden, da die Seite des Abschnitt “Fingering_engraver” in Referenz der Interna schreibt:

Fingering_engraver is part of contexts: ... Abschnitt “Voice” in Referenz der Interna

5.2.4 Benennungskonventionen

Die Bezeichnungen für Funktionen, Variablen, Engraver und Objekte folgen bestimmten Regeln:

- Scheme-Funktionen: kleinbuchstaben-mit-bindestrichen
- Scheme-Funktionen: ly:plus-scheme-stil
- Musikalische Ereignisse, Musikklassen und Musikeigenschaften: wie-scheme-funktionen
- Grob-Schnittstellen: scheme-stil
- backend-Eigenschaften: scheme-stil (aber X und Y)
Kapitel 5: Standardeinstellungen verändern

• Kontexte: Großbuchstabe, oder Großbuchstabe Zwischen Wörtern (CamelCase)
• Kontext-Eigenschaften: kleinbuchstabe Mit Folgenden Großbuchstaben
• Engraver: Großbuchstabe gefolgt von kleinbuchstaben mit unterstrichen

5.3 Eigenschaften verändern

5.3.1 Grundlagen zum Verändern von Eigenschaften


Die \set- und \unset-Befehle werden benutzt, um die Werte von Kontexteigenschaften zu ändern. Die Befehle \override und \revert hingegen verändern die Werte von Grob-Eigenschaften.

Siehe auch


Bekannte Probleme und Warnungen


5.3.2 Der \set-Befehl

Jeder Kontext kann unterschiedliche Eigenschaften besitzen, Variablen, die in diesem Kontext definiert sind. Sie können während der Interpretation des Kontextes verändert werden. Hierzu wird der \set-Befehl eingesetzt:

\set Kontext.Eigenschaft = #Wert

Wert ist ein Scheme-Objekt, weshalb ihm # vorangestellt werden muss.


Mehrtaktpausen etwa können in einen Takt zusammengefasst werden, wenn die Kontexteigenschaft skipBars (Takte überspringen) auf #t (wahr) gesetzt wird:

R1*2
\set Score.skipBars = ##t
R1*2
\begin{music}
\[C-\text{c}\]\end{music}
Wenn das Kontext-Argument ausgelassen wird, bezieht sich der Befehl auf den gerade aktiven unterstößlichen Kontext, üblicherweise ChordNames, Voice oder Lyrics.

\set Score.autoBeaming = ##f
\relative {
  e''8 e e e
  \set autoBeaming = ##t
  e8 e e e
} \"
\relative {
  c''8 c c c c8 c c c
}

Die Änderung wird zur Laufzeit während der Musik interpretiert, sodass diese Einstellung sich nur auf die zweite Gruppe von Achteln auswirkt.

Dabei gilt zu beachten, dass der unterste Kontext nicht immer die Eigenschaft enthält, die verändert werden soll. Wenn man beispielsweise skipBars aus dem oberen Beispiel ohne Angabe des Kontextes zu verändern sucht, hat der Befehl keine Auswirkung, weil er sich auf den Voice-Kontext bezieht, die Eigenschaft sich aber im Score-Kontext befindet:

R1\*2
\set skipBars = ##t
R1\*2

Kontexte sind hierarchisch angeordnet. Wenn ein übergeordneter Kontext angegeben wird, etwa Staff, dann beziehen sich die Änderungen auf alle Stimmen (Voice), die in diesem Kontext enthalten sind.

Es gibt auch einen \unset-Befehl:

\unset Kontext.Eigenschaft
der bewirkt, dass die vorgenommenen Definitionen für Eigenschaft entfernt werden. Dieser Befehl macht nur Einstellungen im richtigen Kontext rückgängig. Wenn also im Staff-Kontext die Bebalkung ausgeschaltet wird:

\set Score.autoBeaming = ##t
\relative {
  \unset autoBeaming
  e''8 e e e
  \unset Score.autoBeaming
  e8 e e e
} \"
\relative {
  c''8 c c c c8 c c c
}

Kontexte sind hierarchisch angeordnet. Wenn ein übergeordneter Kontext angegeben wird, etwa Staff, dann beziehen sich die Änderungen auf alle Stimmen (Voice), die in diesem Kontext enthalten sind.

Es gibt auch einen \unset-Befehl:

\unset Kontext.Eigenschaft
der bewirkt, dass die vorgenommenen Definitionen für Eigenschaft entfernt werden. Dieser Befehl macht nur Einstellungen im richtigen Kontext rückgängig. Wenn also im Staff-Kontext die Bebalkung ausgeschaltet wird:

\set Score.autoBeaming = ##t
\relative {
  \unset autoBeaming
  e''8 e e e
  \unset Score.autoBeaming
  e8 e e e
} \"
\relative {
  c''8 c c c c8 c c c
}
Kapitel 5: Standardeinstellungen verändern

Wie für \set muss das Kontext-Argument für den untersten Kontext nicht mitangegeben werden. Die zwei Versionen
\set Voice.autoBeaming = ##t
\set autoBeaming = ##t

verhalten sich gleich, wenn die gegenwärtige Basis der Voice-Kontext ist.

Einstellungen, die nur einmal vorgenommen werden sollen, können mit \once notiert werden, etwa:

c''4
\once \set fontSize = #4.7
c''4
c''4

Eine vollständige Beschreibung aller vorhandenen Kontexteigenschaften findet sich in der Referenz der Interna, siehe „Translation → Tunable context properties“.

Siehe auch

Internals Reference:
Abschnitt “Tunable context properties” in Referenz der Interna.

5.3.3 Der \override-Befehl


\override ist eigentlich eine Kurzform, der Befehl
\override Kontext.GrobBezeichnung #'Eigenschaft = #Wert

ist äquivalent zu
\set Kontext.GrobBezeichnung =
#(cons (cons 'Eigenschaft Wert)
<vorheriger Wert von Kontext.GrobBezeichnung>)


Beispielsweise kann die Dicke eines Notenhalses verändert werden, indem man die thickness-Eigenschaft des Stem-Objekts verändert:

c4 c
\override Voice.Stem#thickness = #3.0
c4 c
Wenn kein Kontext angegeben wird, wird der tiefste aktuelle Kontext benutzt:

\begin{verbatim}
\override Staff.Stem.thickness = #3.0
<<
  e4 e
  \override Stem.thickness = #0.5
  e4 e
\} \\ {
  c4 c c c
}
\end{verbatim}

Die Auswirkungen von `\override` können mit `\revert` wieder rückgängig gemacht werden:
\begin{verbatim}
c4
\override Voice.Stem.thickness = #3.0
c4 c
\revert Voice.Stem.thickness
c4
\end{verbatim}

Die Auswirkungen von `\override` und `\revert` wirken sich auf alle Grobs im entsprechenden Kontext aber der Stelle aus, an der sie gesetzt werden:
\begin{verbatim}
{ <<
  \{
    e4
    \override Staff.Stem.thickness = #3.0
    e4 e e
  \} \\ {
    c4 c c c
    \revert Staff.Stem.thickness
    c4
  }
  \end{verbatim}

\begin{verbatim}
\end{verbatim}

\begin{verbatim}
/once kann zusammen mit `\override` benutzt werden, um nur den aktuellen Zeitwert zu verändern:
\begin{verbatim}
\end{verbatim}
5.3.4 Der \tweak-Befehl

Wenn man Grob-Eigenschaften mit \override verändert, verändern sich alle fraglichen Objekte zu dem gegebenen musikalischen Moment. Manchmal will man allerdings nur ein Grob verändern, anstatt allen Grobs des aktuellen Kontextes. Das kann mit dem \tweak-Befehl erreicht werden, mit dem man Optimierungen vornehmen kann:

\tweak Layout-Objekt #'grob-eigenschaft #Wert

Die Angabe von Layout-Objekt ist optional. Der \tweak-Befehl wirkt sich auf das musikalische Objekt aus, dass direkt auf Wert folgt.

Eine Einleitung der Syntax und Benutzungen des \tweak-(Optimierungs)-Befehls findet sich in Abschnitt “Optimierungsmethoden” in Handbuch zum Lernen.

Wenn mehrere gleichartige Elemente zum gleichen musikalischen Moment auftreten, kann der \override-Befehl nicht benutzt werden, um nur einen von ihnen zu verändern: hier braucht man den \tweak-Befehl. Elemente, die mehrfach zum gleichen musikalischen Moment auftreten können sind unter Anderem:

- Notenköpfe von Noten innerhalb eines Akkordes
- Artikulationszeichen an einer einzelnen Note
- Bindebögen zwischen Noten eines Akkordes
- Lämmern für rhythmische Verhältnisse (wie Triolen), die zur gleichen Zeit beginnen

In diesem Beispiel wird die Farbe eines Notenkopfes und die Art ein es anderen Notenkopfes innerhalb eines Akkordes verändert:

\begin{verbatim}
<<
\{\n\override Stem.thickness = #3.0
\e4 e e e\n\}\ \{\nc4\n\once \override Stem.thickness = #3.0
\c4 c c\n\}\n>>
\end{verbatim}

Siehe auch

\tweak kann auch benutzt werden, um Bögen zu verändern:
\relative { c'-'\tweak thickness #5 ( d e f) }

\begin{music}
\relative { c'\tweak thickness #5 ( d e f) }
\end{music}

Damit der \tweak-Befehl funktioniert, muss er direkt vor dem Objekt stehen, auf das er sich bezieht. Einen ganzen Akkord kann man nicht mit \tweak verändern, weil der Akkord wie ein Kontainer ist, in dem alle Layoutelemente aus Ereignissen innerhalb von EventChord erstellt werden:
\tweak color #red <c e>4
<\tweak color #red c e>4

\begin{music}
\tweak color #red <c e>4
<\tweak color #red c e>4
\end{music}

Der einfache \tweak-Befehl kann \textit{nicht} eingesetzt werden, um Elemente zu verändern, die nicht direkt aus der Eingabe erstellt werden. Insbesondere Hälse, automatische Balken oder Versetzungszeichen lassen sich nicht beeinflussen, weil diese später durch die Layoutobjekte des Notenkopfs erstellt werden und nicht direkt durch den Quelltext.

Derartige indirekt erstellte Layoutobjekte können mit \tweak verändert werden, indem man die ausführliche Form des Befehls einsetzt:
\tweak Stem.color #red
\tweak Beam.color #green c''8 e''
\tweak <c'' e'' >4 \tweak Accidental.font-size #-3 ges''>4

\begin{music}
\tweak Stem.color #red
\tweak Beam.color #green c''8 e''
\tweak <c'' e'' >4 \tweak Accidental.font-size #-3 ges''>4
\end{music}

\tweak kann auch nicht verwendet werden, um Schlüssel oder Taktarten zu verändern, denn sie werden von dem \tweak-Befehl während der Interpretation durch automatisches Einfügen von zusätzlichen Kontextelementen getrennt.

Mehrere \tweak-Befehle können vor ein Notationselement gesetzt werden und alle werden interpretiert:
\begin{verbatim}
c'
-\tweak style #'dashed-line
-\tweak dash-fraction #0.2
-\tweak thickness #3
-\tweak color #red
\glissando
f''
\end{verbatim}

\begin{music}
\begin{verbatim}
c'
-\tweak style #'dashed-line
-\tweak dash-fraction #0.2
-\tweak thickness #3
-\tweak color #red
\glissando
f''
\end{verbatim}
\end{music}

Der Strom der musikalischen Ereignisse (engl. music stream), der aus dem Quelltext erstellt wird, und zu dem auch die automatisch eingefügten Elemente gehören, kann betrachtet werden, siehe Abschnitt “Musikalische Funktionen darstellen” in \textit{Extending}. Das kann nützlich sein, wenn man herausfinden will, was mit dem \tweak-Befehl verändert werden kann.
Siehe auch
Erweitern: Abschnitt “Musikalische Funktionen darstellen” in Extending.

Bekannte Probleme und Warnungen
Der \tweak-Befehl kann nicht benutzt werden, um die Kontrollpunkte eines von mehreren Bindebögen eines Akkordes zu verändern. Anstelle dessen wird der erste Bogen verändert, der in der Eingabedatei auftritt.

5.3.5 \set versus \override

5.3.6 Alisten verändern
Einige vom Benutzer einstellbare Eigenschaften sind intern als \alists (Assoziative Listen) dargestellt, die Paare von Schlüsseln und Werten speichern. Die Struktur einer Aliste ist:

'((Schlüssel1 . Wert1)
 (Schlüssel2 . Wert2)
 (Schlüssel3 . Wert3)
 ...)

Wenn eine Aliste eine Grob-Eigenschaft oder eine Variable der \paper-Umgebung ist, können ihre Schlüssel einzeln verändert werden, ohne andere Schlüssel zu beeinflussen.


'((basic-distance . 9)
 (minimum-distance . 7)
 (padding . 1)
 (stretchability . 5))

Eine Möglichkeit, die Systemen dichter zueinander zu zwingen, ist es, der Wert des basic-distance-Schlüssels (9) zu verändern, sodass der den gleichen Wert wie minimum-distance (7) hat. Um einen einzelnen Schlüssel zu verändern, wird ein geschachtelter Aufruf benutzt:

% default space between staves
\new PianoStaff <<
 \new Staff { \clef treble c''1 }
 \new Staff { \clef bass c1 }
>>

% reduced space between staves
\new PianoStaff \with {
 % this is the nested declaration
 \override StaffGrouper.staff-staff-spacing.basic-distance = #7
} <<
 \new Staff { \clef treble c''1 }
 \new Staff { \clef bass c1 }
>>
Wenn man diese Art des geschachtelten Aufrufs einsetzt, wird der spezifische Schlüssel (basic-distance im obigen Beispiel) verändert, ohne dass sich andere Wert für die gleiche Eigenschaft ändern würden.

Nun sollen die Systeme so dicht wie möglich gesetzt werden, ohne dass Überlappungen vorkommen. Die einfachste Möglichkeit, das zu tun, wäre es, alle vier Wert auf 0 zu setzen. Man muss jedoch nicht vier Werte definieren, sondern die Eigenschaft kann mit einem Aufruf als Aliste vollständig verändert werden:

```
new PianoStaff \with {
  \override StaffGrouper.staff-staff-spacing =
    #'((basic-distance . 0)
      (minimum-distance . 0)
      (padding . 0)
      (stretchability . 0))
} <<
new Staff { \clef treble c''1 }
new Staff { \clef bass c1 }
>>
```

Dabei sollte beachtet werden, dass alle Schlüssel, die bei dieser Weise des Aufrufs nicht explizit aufgelistet sind, auf den Standardwert gesetzt werden, den sie hätten, wenn sie nicht definiert werden. Im Falle von staff-staff-spacing würden alle nicht genannten Schlüsselwerte auf 0 gesetzt (außer stretchability, welche immer den Wert von space hat, wenn sie nicht definiert ist). Somit sind folgende Aufrufe äquivalent:

```
\override StaffGrouper.staff-staff-spacing =
  #'((basic-distance . 7))

\override StaffGrouper.staff-staff-spacing =
  #'((basic-distance . 7)
    (minimum-distance . 0)
    (padding . 0)
    (stretchability . 7))
```

Eine möglicherweise ungewollte Konsequenz hiervon ist, dass alle Standardwerte, die etwa in einer Initialisierungsdatei zu Beginn einer LilyPond-Partitur geladen werden, nach dem Aufruf rückgängig gemacht werden. Im obigen Beispiel werden die initialisierten Standardwerte für padding und minimum-distance (definiert in scm/define-grobs.scm) auf den Standard
Kapitel 5: Standardeinstellungen verändern

rückgesetzt, den sie uninitialisiert hätten (0 in beiden Fällen). Wenn eine Eigenschaft oder Variable in Form einer Aliste (jeder Größe) definiert wird, werden immer alle Schlüsselwerte auf den uninitialisierten Zustand zurückgesetzt. Es ist also sicherer, geschachtelte Aufrufe zu benutzen, wenn man nicht bewusst alle Werte zurücksetzen will.

Achtung: Geschachtelte Aufrufe funktionieren nicht mit Kontexteigenschaften (wie etwa beamExceptions, keyAlterations, timeSignatureSettings, usw.) Diese Eigenschaften können nur verändert werden, indem man sie vollständig als Alisten umdefiniert.

5.4 Nützliche Konzepte und Eigenschaften

5.4.1 Eingabe-Modi

Die Art, wie die Notation einer Eingabedatei interpretiert wird, hängt vom aktuellen Eingabemodus ab.

Chord (Akkordmodus)

Man erreicht ihn durch den Befehl \chordmode. Hierdurch wird die Eingabe entsprechend der Syntax der Akkordnotation interpretiert, siehe Abschnitt 2.7 [Notation von Akkorden], Seite 387. Akkorde werden als Noten auf einem System dargestellt.

Der Akkordmodus wird auch mit dem Befehl \chords initialisiert. Dadurch wird gleichzeitig ein neuer ChordNames-Kontext erstellt, die Eingabe entsprechend der Syntax der Akkordnotation interpretiert und als Akkordbezeichnungen in einem ChordNames-Kontext dargestellt. Siehe [Akkordbezeichnungen drucken], Seite 393.

Drum (Schlagzeugmodus)

Man erreicht ihn mit dem Befehl \drummode. Die Eingabe wird entsprechend der Syntax der Schlagzeugnotation interpretiert, siehe [Grundlagen der Schlagzeugnotation], Seite 365.

Der Schlagzeugmodus wird auch mit dem Befehl \drums aktiviert. Dadurch wird gleichzeitig ein neuer DrumStaff-Kontext erstellt, die Eingabe entsprechend der Syntax der Schlagzeugnotation interpretiert und als Schlagzeugsymbole auf einem Schlagzeugsystem dargestellt. Siehe [Grundlagen der Schlagzeugnotation], Seite 365.

Figure (Ziffernmodus)

Man erreicht ihn mit dem Befehl \figuremode. Die Eingabe wird entsprechend der Syntax für Generalbass interpretiert, siehe [Eingabe des Generalbass], Seite 402.


Fret/tab (Griffsymbol-/Tabulaturmodus)

Es gibt keinen besonderen Eingabemodus für Griffsymbole und Tabulaturen.

Um Tabulaturen zu erstellen, werden Noten oder Akkorde im Notenmodus notiert und dann in einem TabStaff-Kontext interpretiert, siehe [Standardtabulaturen], Seite 322.


Lyrics (Gesangstextmodus)

Der Gesangstextmodus wird auch durch den Befehl `\addlyrics` aktiviert. Dadurch wird auch ein neuer `Lyrics`-Kontext erstellt und ein impliziter `\lyricsto`-Befehl, der den nachfolgenden Gesangstext mit der vorhergehenden Musik verknüpft.

**Markup (Textbeschriftungsmodus)**

Man erreicht ihn mit dem Befehl `\markup`. Die Eingabe wird entsprechend der Syntax für Textbeschriftung interpretiert, siehe Abschnitt A.10 [Textbeschriftungsbefehle], Seite 646.

**Note (Notenmodus)**

Das ist der Standardmodus. Er kann auch mit dem Befehl `\notemode` gefordert werden. Die Eingabe wird als Tonhöhen, Dauern, Beschriftung usw. interpretiert und als musikalische Notation auf einem Notensystem gesetzt.

Es ist normalerweise nicht nötig, den Notenmodus extra anzugeben, aber es kann in bestimmten Situationen durchaus nützlich sein, etwa wenn man in einem Gesangstext-, Akkord- oder einem anderen Modus arbeitet aber ein Zeichen braucht, das nur im Notenmodus benutzt werden kann.

Um etwa Dynamikzeichen vor die Nummern von unterschiedlichen Strophen zu setzen, muss man den Notenmodus betreten:

```latex
\{ c4 c4 c4 c4 \}
\addlyrics {
  \notemode{ \set stanza = \markup{ \dynamic f 1. } }
  To be sung loudly
}
\addlyrics {
  \notemode{ \set stanza = \markup{ \dynamic p 2. } }
  To be sung quietly
}
```

5.4.2 Richtung und Platzierung


**Richtungseinstellung von Artikulationszeichen**

Standardmäßig sind bestimmte Objekte immer nach oben oder unten ausgerichtet, wie Dynamikzeichen oder Fermaten, während andere Objekte zwischen oben und unten wechseln, was vor allem von der Richtung der Notenhälfte abhängt und etwa Bögen und Akzente betrifft.

Die Standardeinstellungen können verändert werden, indem dem Artikulationszeichen ein Ausrichtungsmarkierer vorangeht. Drei derartige Ausrichtungsmarkierer sind vorhanden: `^` (bedeutet „nach oben“), `_` (bedeutet „nach unten“) bzw. `-` (bedeutet „Standardrichtung“ benutzen)
normalerweise weggelassen werden. In diesem Fall wird - angenommen. Eine Richtungsanweisung ist jedoch immer erforderlich vor

- \texttt{\textbackslash tweak}-Befehlen
- \texttt{\textbackslash markup}-(Textbeschriftungs-)Befehlen
- \texttt{\textbackslash tag}-Befehlen
- Textbeschriftungen in reiner Textform, wie etwa -"string"
- Fingersatzanweisungen: -1
- Abkürzungen von Artikulationen, wie -., ->, --

Ausrichtungsmarkierer haben nur eine Auswirkung auf die nächste Note:

\begin{verbatim}
\relative { 
    c''2( c) 
    c2_( c) 
    c2( c) 
    c2^( c) 
}
\end{verbatim}

Die direction-(Richtungs-)Eigenschaft

Die Position oder Richtung vieler Layoutobjekte wird von der direction-Eigenschaft kontrolliert.

Der Wert der direction-Eigenschaft kann auf den Wert 1 gesetzt werden, was gleichbedeutend mit „nach oben“ bzw. „oberhalb“ ist, oder auf den Wert -1, was „nach unten“ bzw. „unterhalb“ bedeutet. Die Symbole UP und DOWN können anstelle von 1 und -1 benutzt werden. Die Standardausrichtung kann angegeben werden, indem direction auf den Wert 0 oder CENTER gesetzt wird. In vielen Fällen bestehen auch vordefinierte Befehle, mit denen die Ausrichtung bestimmt werden kann. Sie haben die Form \texttt{\textbackslash xxxUp}, \texttt{\textbackslash xxxDown}, \texttt{\textbackslash xxxNeutral}

wobei \texttt{\textbackslash xxxNeutral} bedeutet: „Benutze die Standardausrichtung“. Siehe auch Abschnitt “within-staff (Objekte innerhalb des Notensystems)” in Handbuch zum Lernen.

In wenigen Fällen, von denen Arpeggio das einzige häufiger vorkommende Beispiel darstellt, entscheidet der Wert von direction, ob das Objekt auf der rechten oder linken Seite das Ursprungsobjektes ausgegeben wird. In diesem Fall bedeutet -1 oder LEFT „auf der linken Seite“ und 1 oder RIGHT „auf der rechten Seite“. 0 oder CENTER bedeutet „benutze Standardausrichtung“.

Diese Ausrichtungsanzeigen wirken sich auf alle Noten aus, bis sie rückgängig gemacht werden:

\begin{verbatim}
\relative { 
    c''2( c) 
    \slurDown 
    c2( c) 
    c2( c) 
    \slurNeutral 
    c2( c) 
}
\end{verbatim}
In polyphoner Musik ist es normalerweise besser, eine explizite Stimme (voice) zu erstellen, als die Richtung eines Objektes zu ändern. Zu mehr Information siehe Abschnitt 1.5.2 [Mehrere Stimmen], Seite 157.

Siehe auch

Handbuch zum Lernen: Abschnitt “within-staff (Objekte innerhalb des Notensystems)” in Handbuch zum Lernen.

Notationsreferenz: Abschnitt 1.5.2 [Mehrere Stimmen], Seite 157.

5.4.3 Abstände und Maße

In LilyPond gibt es zwei Arten von Abständen: absolute und skalierte.

Absolute Abstände werden benutzt, um Ränder, Einzüge und andere Einzelheiten des Seitenlayouts zu bestimmen. Sie sind in den Standardeinstellungen in Millimetern definiert. Abstände können auch in anderen Einheiten definiert werden, indem folgende Befehle auf die Zahl folgen: `\mm`, `\cm`, `\in` (Zoll=2,54 cm) und `\pt` (Punkte, 1/72.27 eines Zolls). Abstände des Seitenlayouts können auch in skalierbaren Einheiten (siehe folgenden Absatz) definiert werden, indem man den Befehl `\staff-space` an die Zahl hängt. Das Seitenlayout ist genauer beschrieben in Abschnitt 4.1 [Seitenlayout], Seite 498.


Wenn nur eine ein Abschnitt einer Partitur in einer anderen Größe erscheinen soll, etwa ein Ossia-Abschnitt in einer Fußnote, kann die globale Notensystemgröße nicht einfach geändert werden, weil sich diese Änderung auf die gesamte Partitur auswirken würde. In derartigen Fällen muss die Größenänderung vorgenommen werden, indem man sowohl die `staff-space`-Eigenschaft von `StaffSymbol` als auch die Größe der Schriftarten verändert. Eine Scheme-Funktion, `magstep`, kann von einer Schriftartveränderung zu der entsprechenden Veränderung in `staff-space` (Notenlinienabständen) konvertieren. Zu einer Erklärung und Beispielen zu ihrer Verwendung siehe Abschnitt “Länge und Dicke von Objekten” in Handbuch zum Lernen.

Siehe auch


Notationsreferenz: Abschnitt 4.1 [Seitenlayout], Seite 498, Abschnitt 4.2.2 [Die Notensystemgröße einstellen], Seite 511.

5.4.4 Eigenschaften des Staff-Symbols

Die vertikale Position der Notenlinien und die Anzahl der Notenlinien kann gleichzeitig definiert werden. Wie das folgende Beispiel zeigt, werden Notenpositionen nicht durch die Position der Notenlinien verändert:

\begin{verbatim}
\new Staff \with {
  \override StaffSymbol.line-positions = #'(7 3 0 -4 -6 -7)
}
\relative { a4 e' f b | d1 }
\end{verbatim}


\begin{verbatim}
\new Staff \with {
  \override StaffSymbol.width = #23
}
\relative { a4 e' f b | d1 }
\end{verbatim}

5.4.5 Strecker


Alle Strecker unterstützen das spanner-interface (Strecker-Schnittstelle). Ein paar, insbesondere die, die zwischen zwei Objekten eine gerade Linie ziehen, unterstützen auch das line-spanner-interface (Strecker-Linienschnittstelle).

Das spanner-interface benutzen

Diese Schnittstelle stellt zwei Eigenschaften zur Verfügung, die sich auf mehrere Strecker auswirken:

\textit{Die minimum-length-Eigenschaft}


\begin{verbatim}
a~a
a
% increase the length of the tie
-\tweak minimum-length #5
~a
\end{verbatim}
Kapitel 5: Standardeinstellungen verändern

Die Veränderung kann auch eingesetzt werden, um die Länge von Legato- und Phrasierungsbögen zu verändern:

% default
e' \glissando c''

% not effective alone
\once \override Glissando.minimum-length = #20
e' \glissando c''

% effective only when both overrides are present
Kapitel 5: Standardeinstellungen verändern

Das gilt auch für das Beam-(Balken-)Objekt:

\once \override Beam.minimum-length = #20
\once \override Beam.springs-and-rods = #ly:spanner::set-spacing-rods
e'8 e' e' e'

Die to-barline-Eigenschaft

Die zweite nützliche Eigenschaft des spanner-interface ist to-barline (bis zum Taktstrich). In den Standardeinstellungen ist diese Eigenschaft auf „wahr“ gesetzt, was bedeutet, dass ein Strecker, etwa eine Crescendo-Klammer, der an der ersten Note eines Taktes beendet wird, sich nur bis zum vorhergehenden Taktstrich erstreckt. Wenn die Eigenschaft auf „falsch“ gesetzt wird, erstrecken sich die Strecker entsprechend über die Taktlinie hinüber und enden erst an der entsprechenden Note:

\relative { a' \(< a a a ! a a a \break \override Hairpin.to-barline = \#f a \(< a a a ! a a a }

Diese Eigenschaft wirkt sich nicht auf alle Strecker aus. Im Falle von Legato- oder Phrasierungsbögen etwa hat diese Eigenschaft keinen Effekt. Das gilt auch für alle anderen Streckern, bei denen es nicht sinnvoll wäre, sie an einer Taktlinie abzuschließen.
Das line-spanner-interface benutzen

Objekte, die das line-spanner-interface unterstützen, sind unter Anderem:

- DynamicTextSpanner
- Glissando
- TextSpanner
- TrillSpanner
- VoiceFollower

Die Routine, die das Setzen der Matrizen dieser Strecker hervorruft, ist \ly:line-interface::print. Diese Routine bestimmt die exakte Position der zwei Endpunkte und zeichnet eine Linie zwischen ihnen, in dem erforderlichen Stil. Die Position der zwei Endpunkte des Streckers wird in Echtzeit errechnet, aber es ist möglich, ihre Y-Koordinaten zu verändern. Die Eigenschaften, die angegeben werden müssen, sind zwei Ebenen in der Objekthierarchie tiefer angeordnet, aber die Syntax des \override-Befehls ist ziemlich einfach:

\once \override Glissando.bound-details.left.Y = #3
\once \override Glissando.bound-details.right.Y = #-2

\once \glissando b'
\once \break

Die Einheiten für die Y-Eigenschaft werden in Notenlinienabständen angegeben, wobei die Mittellinie des Notensystems die Null darstellt. Für das Glissando ist der Wert von Y am entsprechenden X-Koordinatenpunkt entsprechend dem Mittelpunkt des Notenkopfes, wenn die Linie bis in die Noten hinein weitergeführt werden würde.

Wenn Y nicht gesetzt wird, wird der Wert aus der vertikalen Position des entsprechenden Anknüpfunktes des Streckers errechnet.

Im Fall eines Zeilenumbruchs werden die Werte der Endpunkte in den Unterlisten left-broken bzw. right-broken von bound-details abgelegt. Zum Beispiel:

\override Glissando.breakable = ##t
\override Glissando.bound-details.right-broken.Y = #-3
\glissando \break

Eine Anzahl weitere Eigenschaft der left- und right-Unterlisten der bound-details-Eigenschaft kann auf gleiche Weise wie Y verändert werden:

Für horizontale Strecker, wie Textstrecker und Trillerstrecker ist sein Wert mit 0 definiert.

**attach-dir**
Das entscheidet, wo die Linie auf der X-Achse beginnt und endet, relativ zum Elternobjekt. Ein Wert -1 (oder LEFT) lässt die Linie an der linken Seite der Noten beginnen/enden, mit der sie verknüpft ist.

**X**
Das ist der absolute X-Koordinatenpunkt des Endpunktes. Der Wert wird normalerweise in Echtzeit errechnet, und ihn zu verändern ist normalerweise nicht nützlich.

**stencil**
Linienstrecker können Symbole am Ende oder zu Anfang des Streckers haben, die in dieser Untereigenschaft definiert werden. Die Eigenschaft ist für interne Benutzung, es wird empfohlen, die Eigenschaft *text* zu benutzen.

**text**
Das ist eine Textbeschriftung, die ausgewertet wird und die *stencil*-Eigenschaft überschreibt. Sie wird eingesetzt, um *cresc.*, *tr* oder andere Texte an horizontale Strecker zu setzen.

```latex
\texttt{\override TextSpanner.bound-details.left.text = \markup { \small \bold Slower }
\relative { c''2\startTextSpan b c a\stopTextSpan }}
```

**stencil-align-dir-y**

**stencil-offset**
Wenn keine dieser beiden Eigenschaften gesetzt wird, wird die Matrise (engl. *stencil*) einfach am Endpunkt des Streckers, auf seiner Mittellinie (wie durch *X* und *Y* definiert) zentriert, ausgegeben. Wenn entweder *stencil-align-dir-y* oder *stencil-offset* gesetzt werden, wird das Symbol am Rand vertikal entsprechend des Endpunktes der Linie verschoben:

```latex
\texttt{\override TextSpanner.bound-details.left.stencil-align-dir-y = #-2
\override TextSpanner.bound-details.right.stencil-align-dir-y = #UP}
```

```latex
\texttt{\relative { c'4\startTextSpan c c c \stopTextSpan } 
\texttt{\override TextSpanner.bound-details.left.text = "ggg"
\override TextSpanner.bound-details.right.text = "hhh"}
```

Dabei sollte beachtet werden, dass negative Werte das Objekt nach *oben* verschieben, anders als man erwarten könnte, weil der Wert -1 oder DOWN bedeutet, dass die Unterkante des Textes mit der Streckerlinie ausgerichtet wird. Ein Wert 1 oder UP richtet die Oberkante des Textes mit der Streckerlinie aus.

**arrow**
Wenn diese Untereigenschaft auf #t gesetzt wird, wird ein Pfeilkopf am Ende der Linie erstellt.

Die musikalische Funktion `\endSpanners` beschließt den Strecker, der an der direkt folgenden Note beginnt, bevor er eigentlich zu ende wäre. Er wird exakt nach einer Note beendet, oder am nächsten Taktstrich, wenn `to-barline` auf wahr gesetzt ist und eine Taktlinie vor der nächsten Note erscheint.

\begin{verbatim}
relative c' { 
endSpanners
  c2 \startTextSpan c2 c2
  \endSpanners
  c2 \< c2 c2 
}
\end{verbatim}

Wenn man `\endSpanners` benutzt, ist es nicht nötig, den Befehl `\startTextSpan` mit `\stopTextSpan` zu beenden, und es ist auch nicht nötig, Crescendo-Klammern mit `!` zu beenden.

Siehe auch


5.4.6 Sichtbarkeit von Objekten


Es gibt auch einige weitere Techniken, die sich nur auf bestimmte Layout-Objekte beziehen. Sie werden im letzten Abschnitt behandelt.

Einen stencil entfernen

Jedes Layout-Objekt hat eine Matrizen-(stencil)-Eigenschaft. Sie ist normalerweise definiert als die Funktion, die das entsprechende Objekt zeichnet. Wenn die Eigenschaft mit `\override` auf `#f` gesetzt wird, wird keine Funktion aufgerufen und also auch kein Objekt gezeichnet. Das Standardverhalten kann mit dem Befehl `\revert` wieder hergestellt werden.
Objekten unsichtbar machen

Jedes Layout-Objekt hat eine Durchsichtigkeits-Eigenschaft (\texttt{transparent}), die normalerweise auf den Wert \texttt{#f} gesetzt ist. Wenn sie auf \texttt{#t} gesetzt wird, nimmt das Objekt immer noch den entsprechenden Platz ein, ist aber unsichtbar.

```
\once \hide NoteHead
```

Objekte weiß malen

Alle Layout-Objekte haben eine Farb-(color)-Eigenschaft, die normalerweise schwarz (\texttt{black}) definiert ist. Wenn sie nach weiß (\texttt{white}) verändert wird, kann man das Objekt nicht mehr vom weißen Hintergrund unterscheiden. Wenn das Objekt jedoch andere Objekte überschneidet, wird die Farbe der Überschneidungen von der Reihenfolge entschieden, in welcher die Objekte gesetzt werden. Es kann also vorkommen, dass man die Umrisse des weißen Objektes erahnen kann, wie in diesem Beispiel:

```
\override Staff.Clef.color = #white
```

Das kann man vermeiden, indem man die Satzreihenfolge der Objekte verändert. Alle Layout-Objekte haben eine \texttt{layer}-Eigenschaft, die auf eine ganze Zahl gesetzt sein muss. Objekte mit der niedrigsten Zahl in der \texttt{layer}-Eigenschaft werden zuerst gesetzt, dann die nächsten Objekte in ansteigender Ordnung. Objekte mit höheren Werten überschneiden also Objekte mit niedrigeren Werten. Die meisten Objekte bekommen den Wert 1 zugewiesen, einige wenige Objekte, unter die auch \texttt{StaffSymbol} (die Notenlinien) gehört, jedoch den Wert 0. Die Reihenfolge, in der Objekte mit demselben Wert gesetzt werden, ist nicht definiert.

Im oberen Beispiel wird der weiße Schlüssel, der einen Wert von 1 für \texttt{layer} hat, nach den Notenlinien gesetzt (die einen Wert von 0 für \texttt{layer} haben) und überschneidet sie also. Um das zu ändern, muss dem \texttt{Clef}-Objekt (Notenschlüssel) ein niedrigerer Wert, etwa \texttt{-1}, gegeben werden, sodass es früher gesetzt wird:

```
\override Staff.Clef.color = #white
\override Staff.Clef.layer = #-1
```

\begin{figure}
\centering
\includegraphics[width=0.4\textwidth]{example.png}
\end{figure}

\textbf{break-visibility (unsichtbar machen) benutzen}

Die meisten Layout-Objekte werden nur einmal gesetzt, aber einige, wie Taktstriche, Schlüssel, Taktartbezeichnung und Tonartvorzeichen, müssen mehrmals gesetzt werden, wenn die Zeile gewechselt wird: einmal am Ende des oberen Systems und ein zweites Mal zu Beginn des nächsten Systems. Derartige Objekte werden als \textit{trennbar} bezeichnet und haben eine Eigenschaft, die
break-visibility-Eigenschaft, mit der ihre Sichtbarkeit an allen drei Positionen, an denen sie auftreten können, kontrolliert werden kann: zu Beginn einer Zeile, innerhalb einer Zeile, wenn sie verändert werden, und am Ende einer Zeile, wenn die Änderung hier stattfindet.

Die Taktart wird beispielsweise standardmäßig nur zu Beginn des ersten Systems gesetzt, aber an anderen Stellen nur, wenn sie sich ändert. Wenn diese Änderung am Ende eines Systems auftritt, wird die neue Taktart am Ende des aktuellen Systems als auch zu Beginn des nächsten Systems gesetzt.


Die acht möglichen Kombinationen können auch durch vordefinierte Funktionen bestimmt werden, welche in der Datei scm/output-lib.scm definiert sind. Die letzten drei Spalten der folgenden Tabelle zeigen an, ob das Layout-Objekt an einer bestimmten Position sichtbar sein wird oder nicht:

<table>
<thead>
<tr>
<th>Funktion Form</th>
<th>Vektor Form</th>
<th>Vor Umbruch</th>
<th>Kein Umbruch</th>
<th>Nach Umbruch</th>
</tr>
</thead>
<tbody>
<tr>
<td>all-visible</td>
<td>'t t t</td>
<td>ja</td>
<td>ja</td>
<td>ja</td>
</tr>
<tr>
<td>begin-of-line-visible</td>
<td>'f f t</td>
<td>nein</td>
<td>nein</td>
<td>ja</td>
</tr>
<tr>
<td>center-visible</td>
<td>'f t f</td>
<td>nein</td>
<td>ja</td>
<td>nein</td>
</tr>
<tr>
<td>end-of-line-visible</td>
<td>'t f f</td>
<td>ja</td>
<td>nein</td>
<td>nein</td>
</tr>
<tr>
<td>begin-of-line-invisible</td>
<td>'t t f</td>
<td>ja</td>
<td>ja</td>
<td>nein</td>
</tr>
<tr>
<td>center-invisible</td>
<td>'t f f</td>
<td>ja</td>
<td>nein</td>
<td>ja</td>
</tr>
<tr>
<td>end-of-line-invisible</td>
<td>'f f t</td>
<td>nein</td>
<td>ja</td>
<td>ja</td>
</tr>
<tr>
<td>all-invisible</td>
<td>'f f f</td>
<td>nein</td>
<td>nein</td>
<td>nein</td>
</tr>
</tbody>
</table>

Die Standardeinstellungen von break-visibility hängen vom Layout-Objekt ab. Die folgende Tabelle zeigt alle wichtigen Layout-Objekte, die mit break-visibility verändert werden können und die jeweiligen Standardeinstellungen der Eigenschaft:

<table>
<thead>
<tr>
<th>Layout-Objekt</th>
<th>Normaler Kontext</th>
<th>Standardeinstellung</th>
</tr>
</thead>
<tbody>
<tr>
<td>BarLine (Taktstrich)</td>
<td>Score</td>
<td>calculated</td>
</tr>
<tr>
<td>BarNumber (Taktzahl)</td>
<td>Score</td>
<td>begin-of-line-visible</td>
</tr>
<tr>
<td>BreathingSign (Atemzeichen)</td>
<td>Voice</td>
<td>begin-of-line-visible</td>
</tr>
<tr>
<td>Clef (Schlüssel)</td>
<td>Staff</td>
<td>begin-of-line-visible</td>
</tr>
<tr>
<td>Custos</td>
<td>Staff</td>
<td>end-of-line-visible</td>
</tr>
<tr>
<td>DoublePercentRepeat (Doppel-Prozent-Wiederholung)</td>
<td>Voice</td>
<td>begin-of-line-visible</td>
</tr>
<tr>
<td>KeySignature (Tonart)</td>
<td>Staff</td>
<td>begin-of-line-visible</td>
</tr>
<tr>
<td>ClefModifier (Oktavierungs-Acht)</td>
<td>Staff</td>
<td>begin-of-line-visible</td>
</tr>
<tr>
<td>RehearsalMark (Übungszeichen)</td>
<td>Score</td>
<td>end-of-line-invisible</td>
</tr>
</tbody>
</table>
Kapitel 5: Standardeinstellungen verndern

TimeSignature (Taktart)  Staff  all-visible

Das Beispiel unten zeigt die Verwendung der Vektor-Form um die Sichtbarkeit von Taktlinien zu bestimmen:

\once \override Score.BarLine.break-visibility = ##(#f #t #t)
\break
f4 g a b
f4 g a b
% Remove bar line at the end of the current line
\relative {
  \key g \major
  f'4 g a b
  % Try to remove all key signatures
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b
  \break
  f4 g a b
}

Obwohl alle drei Bestandteile des Vektors, mit denen break-visibility definiert wird, vorhanden sein müssen, haben nicht alle eine Auswirkung auf jedes Layout-Objekt, und einige Kombinationen können sogar Fehler hervorrufen. Es gelten die folgenden Einschränkungen:

- Taktstriche können nicht zu Beginn einer Zeile gesetzt werden.
- Eine Taktzahl kann nicht zu Beginn der ersten Zeile gesetzt werden, außer wenn er nicht 1 ist.
- Schlüssel – siehe unten.
- Doppel-Prozent-Wiederholungen werden entweder alle ausgegeben oder alle unterdrückt. Mit begin-of-line-invisible werden sie ausgegeben, mit all-invisible unterdrückt.
- Tonart – siehe unten.
- Oktavierungs-Acht – siehe unten.

**Besonderheiten**

*Sichtbarkeit nach expliziten Änderungen*

Die break-visibility-Eigenschaft kontrolliert die Sichtbarkeit von Tonarten und Schlüsseländerungen nur zu Beginn einer Zeile, d.h. nach einem Zeilenumbruch. Sie hat keinen Einfluss auf die Sichtbarkeit von Tonarten bzw. Schlüsseln, die nach einer expliziten Tonart- oder Schlüsseländerung in oder am Ende einer Zeile angezeigt werden. Im nächsten Beispiel ist die Tonartangabe nach dem expliziten Wechsel zu B-Dur immer noch sichtbar, obwohl all-invisible eingesetzt wurde:

\relative {
  \key g \major
  f'4 g a b
  % Try to remove all key signatures
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b
  \break
  f4 g a b
}

```latex
\relative {
\key g \major
f'4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Staff.KeySignature.break-visibility = #all-invisible
\key bes \major
f4 g a b \break
f4 g a b
f4 g a b
}
```

*Sichtbarkeit von erinnernden Versetzungszeichen*

Um erinnernde Versetzungszeichen zu entfernen, die nach einer expliziten Tonartänderung auftreten, muss die `Staff`-Eigenschaft `printKeyCancellation` auf `#f` gesetzt werden:

```latex
\relative {
\key g \major
f'4 g a b
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.printKeyCancellation = ##f
\override Staff.KeySignature.break-visibility = #all-invisible
\key bes \major
f4 g a b \break
```
Mit diesen Veränderungen bleiben nur noch die Versetzungszeichen vor den Noten übrig um den Wechsel der Tonart anzuzeigen.

**Automatische Takte**

Ein Sonderfall sind die automatischen Taktstriche, die mit der Eigenschaft `automaticBars` im Score-Kontext ausgeschaltet werden können. Wenn sie auf `#f` gesetzt ist, werden Taktstriche nicht automatisch ausgegeben sondern müssen explizit mit dem `\bar`-Befehl eingegeben werden. Anders als bei dem `\cadenzaOn`-Befehl werden die Takte allerdings immer noch gezählt. Takterstellung wird später wieder mit diesem Zahl aufgenommen, wenn die Eigenschaft wieder auf `#t` gesetzt wird. Wenn sie den Wert `#f` hat, können Zeilenumbrüche nur an expliziten `\bar`-Befehlen auftreten.

**Oktavierte Schlüssel**


Bei expliziten Schlüsseländerungen kontrolliert die `explicitClefVisibility`-Eigenschaft wohl das Schlüsselsymbol als auch das damit verknüpfte Oktavierungssymbol.

**Siehe auch**


**5.4.7 Linienstile**

Einige Aufführungsanweisungen (z. B. `rallentando` und `accelerando` oder Triller werden als Text gesetzt und möglicherweise über mehrere Takte mit Linien fortgeführt, die teilweise gestrichelt oder gewellt sind.

Alle benutzen die gleichen Routinen wie das Glissando, um Text und Linien zu produzieren, weshalb auch eine Veränderungen der Erscheinung auf gleiche Weise vonstatten geht. Die Ausgabe erfolgt durch einen Strecker (engl. spanner), und die Routine, die ihn setzt, heißt `ly:line-interface::print`. Diese Routine bestimmt die exakte Position von zwei Strecker-Punkten und zeichnet eine Linie zwischen sie im gewünschten Linienstil.

Hier einige Beispiele, welche Linienstile möglich sind und wie sie verändert werden können:

```
\relative {
  \once \override Glissando.style = #'dashed-line
  d,2 \glissando d'2
  \override Glissando.style = #'dotted-line
```
Die Position der Endpunkte des Streckers werden in Realzeit für jedes graphische Objekt errechnet, aber es ist möglich, sie manuell vorzugeben:

\relative {
    e''2 \glissando f
    \once \override Glissando.bound-details.right.Y = #-2
    e2 \glissando f
}
Kapitel 5: Standardeinstellungen verändern

Textbeschriftung drehen

Jede Textbeschriftung kann gedreht werden, indem vor die Anweisung der Befehl \rotate gesetzt wird. Der Befehl hat zwei Argumente: Den Winkel der Drehung in Grad gegen den Uhrzeiger und den Text, der gedreht dargestellt werden soll. Die Ausdehnung des Textes wird nicht gedreht, sie erhält ihren Wert von den Extrempunkten der x- und y-Koordinaten des gedrehten Textes. Im folgenden Beispiel wird die outside-staff-priority-Eigenschaft auf ##f gesetzt, damit automatische Zusammenstöße nicht verhindert werden, wodurch andernfalls einige der Texte zu hoch geschoben werden würden.

\override TextScript.outside-staff-priority = ##f
g4`\markup { \rotate #30 "a G" }
b`\markup { \rotate #30 "a B" }
des`\markup { \rotate #30 "a D-Flat" }
fis`\markup { \rotate #30 "an F-Sharp" }

5.5 Fortgeschrittene Optimierungen

Dieser Abschnitt behandelt verschiedene Möglichkeiten, das Aussehen des Notenbildes zu polieren.

Siehe auch


Notationsreferenz: Abschnitt 5.2 [Die Referenz der Programminterna erklärt], Seite 568, Abschnitt 5.3 [Eigenschaften verändern], Seite 572.


Installierte Dateien: scm/define-grobs.scm.


Referenz der Interna: Abschnitt “All layout objects” in Referenz der Interna.

5.5.1 Objekte ausrichten


\begin{itemize}
\item \textbf{Achtung:} Viele Objekte brauchen besondere Überlegungen zu ihrer Position, weshalb in manchen Fällen manuell gesetzte Werte von \textit{X-offset} oder \textit{Y-offset} ignoriert oder verändert werden können, obwohl das Objekt das \textit{self-alignment-interface} unterstützt. Wenn man \textit{X-offset} oder \textit{Y-offset} auf einen festen Wert setzt, wird die entsprechende \textit{self-alignment}-Eigenschaft ignoriert.
\end{itemize}

Ein Versetzungssymbol beispielsweise kann vertikal durch Veränderung von \textit{Y-offset} verschoben werden, aber Änderungen von \textit{X-offset} haben keine Auswirkung.


\section*{Siehe auch}

- Notationshandbuch: [Benutzung des break-alignable-interface], Seite 599.
- Erweitern: Abschnitt "Callback functions" in \textit{Extending}.

\section*{\textit{X-offset} und \textit{Y-offset} direkt setzen}


\begin{verbatim}
\tweak X-offset #0
\tweak Y-offset #0
-3
a'
\tweak X-offset #-1
\tweak Y-offset #1
-3
\end{verbatim}

\begin{figure}[h]
\centering
\includegraphics[width=0.2\textwidth]{example.png}
\caption{Example of \textit{X-offset} and \textit{Y-offset} settings.}
\end{figure}

\section*{Das side-position-interface benutzen}


Die Werte von \textit{side-axis} und \textit{direction} bestimmen, wo das Objekt platziert werden soll, wie in der Tabelle zu sehen:
Kapitel 5: Standardeinstellungen verändern

side-axis-Eigenschaft | direction-Eigenschaft | Platzierung
---|---|---
0 | -1 | links
0 | 1 | rechts
1 | -1 | unten
1 | 1 | oben


Das self-alignment-interface benutzen

Selbstausrichtende Objekte horizontal


Normalerweise würde der \override-Befehl benutzt werden, um die Werte von self-alignment-X zu verändern, aber der \tweak-Befehl kann benutzt, um verschiedene Anmerkungen an einer einzigen Note auszurichten:

\override a'  
\tweak self-alignment-X #-1  
\"left-aligned\"  
\tweak self-alignment-X #0  
\"center-aligned\"  
\tweak self-alignment-X #RIGHT  
\"right-aligned\"  
\tweak self-alignment-X #-2.5  
\"aligned further to the right\"

right-aligned  
center-aligned  
left-aligned  
aligned further to the right

Objekte vertikal automatisch ausrichten

*Automatische Ausrichtung in beide Richtungen*

Indem sowohl X-offset als auch Y-offset eingestellt werden, kann ein Objekt gleichzeitig in beiden Richtungen ausgerichtet werden.

Das folgende Beispiel zeigt, wie man eine Fingersatzanweisung so ausrichtet, dass sie nah am Notenkopf bleibt.

```latex
a'
-\tweak self-alignment-X #0.5 % move horizontally left
-\tweak Y-offset #ly:self-alignment-interface::y-aligned-on-self
-\tweak self-alignment-Y #1 % move vertically up
-3 % third finger
```

**Benutzung des break-alignable-interface**

Übungszeichen und Taktzahlen können an Notationsobjekten (ausschließlich Taktstriche) ausgerichtet werden. Zu diesen Objekten gehören `ambitus`, `breathing-sign`, `clef`, `custos`, `staff-bar`, `left-edge`, `key-cancellation`, `key-signature` und `time-signature`.

Standardmäßig werden Übungszeichen und Taktzahlen horizontal über dem Objekt zentriert:

```latex
% The RehearsalMark will be centered above the Clef
\override Score.RehearsalMark.break-align-symbols = #'(clef)
\key a \major
\clef treble
\mark "↓"
e1
% The RehearsalMark will be centered above the TimeSignature
\override Score.RehearsalMark.break-align-symbols = #'(time-signature)
\key a \major
\clef treble
\time 3/4
\mark "↓"
e2.
% The rehearsal mark will be centered above the Breath Mark
\override Score.RehearsalMark.break-align-symbols = #'(breathing-sign)
\key a \major
\clef treble
\time 4/4
\mark "↓"
e1
```
Eine Liste von möglichen Objekten zur Ausrichtung kann definiert werden. Wenn eins dieser Objekte an der aktuellen Stelle unsichtbar ist (etwa durch Einstellung von \texttt{break-visibility} oder die expliziten Sichtbarkeitseinstellungen von Taktart und Vorzeichen), werden Übungszeichen und Taktzahlen an dem ersten Objekt in der Liste ausgerichtet, dass sichtbar ist. Wenn keine Objekte in der Liste sichtbar sind, wird das Objekt am Taktstrich ausgerichtet. Wenn der Taktstrich unsichtbar ist, wird das Objekt an der Stelle ausgerichtet, an der sich der Taktstrich befinden würde.

\begin{verbatim}
% The RehearsalMark will be centered above the Key Signature
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark "↓"
\e1
% The RehearsalMark will be centered above the Clef
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef bass
\mark "↓"
\gis,,1
% The rehearsal mark will be centered above the Bar Line
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.explicitClefVisibility = #all-invisible
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark "↓"
\e''1
\end{verbatim}


\begin{verbatim}
% The RehearsalMark will be centered above the KeySignature
\override Score.RehearsalMark.break-align-symbols = #'(key-signature)
\key a \major
\clef treble
\time 4/4
\mark "↓"
\end{verbatim}
Das Übungszeichen kann auch nach rechts oder links um einen beliebigen Wert verschoben werden. Die Einheiten sind in Notenlinienzwischenräumen:


5.5.3 stencils verändern

Alle Layout-Objekte haben eine `stencil`-(Stempel-)Eigenschaft, die ein Teil von `grob-interface` ist. Diese Eigenschaft ist normalerweise als eine Funktion definiert, die auf das jeweilige Objekt angepasst ist und das Symbol erstellt, dass dann im Druckbild erscheint. Beispielsweise die Standardeinstellung für die `stencil`-Eigenschaft von `MultiMeasureRest` (Ganztaktpausenobjekt) ist `ly:multi-measure-rest::print`. 
Das Standardsymbol für jedes Objekt kann ersetzt werden, indem man die stencil-Eigenschaft verändert, sodass sie auf eine andere, speziell geschriebene Prozedur verweist. Das erfordert einen hohen Grad an Kenntnis der LilyPond-Interna, aber es gibt einen einfacheren Weg, mit dem oft vergleichbarere Ergebnisse erzielen kann.

Dieser Weg besteht darin, die stencil-Eigenschaft auf die Prozedur zu verweisen, die Text ausgibt: ly:text-interface::print und eine text-Eigenschaft zu dem Objekt hinzuzufügen, in welcher dann die Textbeschriftung definiert wird, mit der das entsprechende Symbol dargestellt wird. Aufgrund der Flexibilität der Textbeschriftung ist hier sehr viel möglich. Siehe zu Details insbesondere [Graphische Notation innerhalb einer Textbeschriftung], Seite 231.

Das folgende Beispiel zeigt diese Methode, indem das Symbol der Notenkopfe in ein Kreuz innerhalb eines Kreises umgewandelt wird.

\begin{verbatim}
Xin0 = {
  \once \override NoteHead.stencil = #ly:text-interface::print
  \once \override NoteHead.text = \markup { 
  \combine 
  \halign #-0.7 \draw-circle #0.85 #0.2 ##f
  \musicglyph "noteheads.s2cross"
  }
}
\relative {
  a' a \Xin0 a a
}
\noteset {4 4/noteset-s2}
\end{verbatim}

Alle Schriftzeichen in der Feta-Glyphe können mit dem \musicglyph-Befehl erreicht werden. Siehe auch Abschnitt A.8 [Die Emmentaler-Schriftart], Seite 625.

Siehe auch

Notationsreferenz: [Graphische Notation innerhalb einer Textbeschriftung], Seite 231, Abschnitt 1.8.2 [Text formatieren], Seite 223, Abschnitt A.10 [Textbeschriftungsbefehle], Seite 646, Abschnitt A.8 [Die Emmentaler-Schriftart], Seite 625.

5.5.4 Formen verändern

Bögen verändern

Binde-, Legato- und Phrasierungsbögen werden als Bézierkurven dritter Ordnung gezeichnet. Wenn die Form eines automatischen Bogens nicht optimal ist, kann sie manuell verändert werden, indem man die vier erforderlichen Kontrollpunkte angibt.

Bézierkurven dritter Ordnung (auch als quadratische Bézierkurven bezeichnet) werden durch vier Kontrollpunkte definiert. Der erste und vierte Kontrollpunkt geben Beginn und Ende der Kurve an. Die zwei Punkte dazwischen werden benutzt, um die Form der Kurve zu bestimmen. Im Internet gibt es Animationen, die illustrieren, wie eine derartige Kurve gezeichnet wird, aber die folgende Beschreibung kann hilfreich sein. Die Kurve beginnt am ersten Kontrollpunkt in Richtung des zweiten, wobei sie sich schrittweise krümmt um zum dritten Kontrollpunkt zu gelangen, von wo aus sie sich weiter zum vierten Punkt hin krümmt. Die Form der Kurve wird vollständig von den vier Punkten definiert.

Hier ein Beispiel eines Falles, in dem der Bogen nicht optimal erscheint, und wo auch \tieDown das Problem nicht lösen würde.
Eine Möglichkeit, diesen Bogen zu verbessern, ist es, seine Kontrollpunkte manuell zu verändern:


Es lohnt sich daran zu denken, dass eine symmetrische Kurve symmetrische Kontrollpunkte benötigt, und dass Bézierkurven die nützliche Eigenschaft haben, dass eine Transformation der Kurve wie eine Übersetzung, Drehung oder Skalierung der Kurve erreicht werden kann, indem man die gleiche Skalierung auf die Kontrollpunkte anwendet.

In dem obigen Beispiel geben folgende Werte einen zufriedenstellenden Bogen – Achtung: der Befehl muss direkt vor dem Beginn der Note gesetzt werden, an die der (Binde-)Bogen angehängt wird.

```
<<
    { e'1~ 1 }
\
    \relative { r4 <g' c,> <g c,> <g c,> }
>>
```

Bekannte Probleme und Warnungen

Es ist nicht möglich, die Form von Bögen anhand ihrer control-points-Eigenschaft zu verändern, wenn mehrere Binde- oder Legatobögen zum gleichen musikalischen Moment auftreten, nicht einmal mit dem \tweak-Befehl. Die Eigenschaft tie-configuration von TieColumn kann jedoch verändert werden, sodass Startlinie und Richtung wie benötigt platziert werden.

Siehe auch

Referenz der Interna: Abschnitt “TieColumn” in Referenz der Interna.
5.5.5 Reine und unreine Container

Unreine und reine Container (engl. unpure/pure containers) sind nützlich, wenn man die Berechnungen der Platzierungen für die Y-Achse verändern will, insbesondere für Y-offset und Y-extent. Mit diesen Containern kann die Veränderung durch eine Scheme-Funktion anstelle einer direkten Zahl oder eines Paares vorgenommen werden.

Für bestimmte Grobs basiert die Eigenschaft Y-extent auf der stencil-Eigenschaft. Wenn diese mit \override verändert werden soll, braucht man eine zusätzliche Veränderung von Y-extent mit einem unreinen-reinen Container. Wenn eine Funktion Y-offset und/oder Y-extent verändert, wird angenommen, dass dadurch Zeilenumbruchsbe rechnungen zu früh während der Kompilierung aufgerufen werden. Die Funktion wird also überhaupt nicht ausgewertet (und gibt also normalerweise den Wert '0' oder '((0 . 0)' zurück), wodurch sich Zusammenstöße ergeben können. Eine „saubere“ Funktion beeinflusst keine Eigenschaften, Objekte oder Grob-Suizide, weshalb ihre Werte, die sich auf Y-axis beziehen, richtig berechnet werden.


Achtung: Da es schwierig ist, immer sicher zu sein, welche Funktionen sich auf dieser Liste befinden, wird empfohlen, dass die selbsterstellten „sauberen“ Funktionen nicht die Grobs Beam oder VerticalAlignment einsetzen.

Ein unsauberer-sauberer Container wird wie folgend erstellt:

(ly:make-unpure-pure-container f0 f1)

wobei f0 eine Funktion ist, die n Arguments braucht (n >= 1) und deren erstes Argument immer der Grob sein muss. Das ist die Funktion, die das eigentliche Resultat ausgibt. f1 ist die Funktion, die als „sauber“ bezeichnet wird, und braucht n + 2 Argumente. Wiederum muss das erste Argument immer der Grob sein, aber das erste und zweite Argument sind „Beginn“ und „Endeargumente“.

start (Beginn) und end (Ende) sind absichtlich nur Platzhalter, die nur für die Strecke gelten (etwa Hairpin oder Beam), die unterschiedliche Höhenberechnungen je nach beginnender und endender Note ausgeben können.

Der Rest sind andere Argumente für die erste Funktion (es können auch Null sein, wenn n = 1).

Die Ergebnisse der zweiten Funktion werden als Näherungswert des benötigten Wertes benutzt, welche dann von der ersten Funktion eingesetzt wird, um den wirklichen Wert auszugeben, mit dem dann sehr viel später im Layoutprozess die Platzierung justiert werden soll.

#(define (square-line-circle-space grob)
  (let* ((pitch (ly:event-property (ly:grob-property grob 'cause) 'pitch))
         (notename (ly:pitch-notename pitch)))
    (if (= 0 (modulo notename 2))
      (make-circle-stencil 0.5 0.0 #t)
      (make-filled-box-stencil '((0 . 1.0)
                               '(-0.5 . 0.5))))

squareLineCircleSpace = {
  \override NoteHead.stencil = #square-line-circle-space
}
Im ersten Takt weiß die Layoutmaschine ohne den unsauberen-sauberen Container nicht die Breite des Notenkopfes und lässt ihn deshalb mit den Versetzungssymbolen zusammenstoßen. Im zweiten Takt, mit unreinen-reinen Containern, weiß die Layoutmaschine die Breite des Notenkopfes und vermeidet den Zusammenstoß, indem die Zeile entsprechend verlängert wird.

Normalerweise können für eine einfache Berechnung fast identische Funktionen für den „unsauberen“ und „sauberen“ Teil benutzt werden, indem nur die Zahl der Argumente und die Reichweite der Funktion verändert wird.

Achtung: Wenn eine Funktion als „sauber“ bezeichnet ist und das aber nicht ist, können unerwartete Ergebnisse auftreten.

5.6 Musikfunktionen benutzen


5.6.1 Syntax der Ersetzungsfunktion

Es ist einfach eine Funktion zu erstellen, die eine Variable in LilyPond-Code umwandelt. Die generelle Form dieser Funktionen ist:

```lilypond
Funktion =
#(define-music-function
 (Arg1 Arg2 ...)
  (Typ1? Typ2? ...)
#{
  ...Noten...
 #})
```

wobei
Kapitel 5: Standardeinstellungen verändern

$ArgN$ ntes Argument

Typ? ein Scheme Typenprädikat, für das $ArgN$ den Wert $\#t$ ausgibt.

...Noten... normale LilyPond-Eingabe, wobei $\$$ (wenn nur LilyPond-Konstruktionen erlaubt sind) oder $\#$ (um es als Scheme-Wert oder Argument einer musikalischen Funktionen oder als Noten innerhalb von Notenlisten einzusetzen) benutzt wird, um Argumente zu referenzieren (etwa $\$Arg1$).

Die parser und location-Argumente sind zwingend und werden in einigen fortgeschrittenen Situationen benutzt, wie sie im „Erweitern“-Handbuch beschrieben werden (siehe Abschnitt “Musikalische Funktionen” in Extending). In Ersetzungsfunktionen gehen Sie einfach sicher, dass sie die beiden Wörter auch mit aufnehmen.

Die Liste der Typenprädikate ist auch notwendig. Einige der häufigsten Typenprädikate, die in musikalischen Funktionen benutzt werden, sind:

- boolean?
- cheap-list? (benutze anstelle von ,list¿ für schnelleres Kompilieren)
- ly:duration?
- ly:music?
- ly:pitch?
- markup?
- number?
- pair?
- string?
- symbol?

Eine Liste aller Typprä dikate findet sich unter Abschnitt A.20 [Vordefinierte Typprä dikate], Seite 756. Eigene Typprä dikate sind auch erlaubt.

Siehe auch

Notationsreferenz: Abschnitt A.20 [Vordefinierte Typprä dikate], Seite 756.
Erweitern: Abschnitt “Musikalische Funktionen” in Extending.
Installierte Dateien: lily/music-scheme.cc, scm/c++.scm, scm/lily.scm.

5.6.2 Beispiele der Ersetzungsfunktionen


Im ersten Beispiel wird eine Funktion definiert, die das Verschieben von TextScript erleichtert:

```scheme
padText =
  #(define-music-function
   (padding)
     (number?)
   #{
     \once \override TextScript.padding = #padding
   };
\relative {
  c''4"piu mosso" b a b
  \padText #1.8
  c4"piu mosso" d e f
```

Neben Zahlen können auch musikalische Ausdrücke wie Noten als Argumente für musikalische Funktionen eingesetzt werden:

```plaintext
custosNote =
#(define-music-function
 (note)
 (ly:music?)
#{
 \tweak NoteHead.stencil #ly:text-interface::print
 \tweak NoteHead.text
 \markup \musicglyph "custodes.mensural.u0"
 \tweak Stem.stencil ##f
 #note
 #})
```

Ersetzungsfunktionen mit mehrfachen Argumenten können definiert werden:

```plaintext
tempoPadded =
#(define-music-function
 (padding tempotext)
 (number? markup?)
#{
 \once \override Score.MetronomeMark.padding = #padding
 \tempo \markup { \bold #tempotext }
 #})
```

```plaintext
\relative { c'4 d e f \custosNote g }
```

Ersetzungsfunktionen mit mehrfachen Argumenten können definiert werden:

```plaintext
\relative {
 \tempo \markup { "Low tempo" }
 c''4 d e f g1
 \tempoPadded #4.0 "High tempo"
 g4 f e d c1
}
```

\relative { c'4 d e f \custosNote g }

\relative {
 \tempo \markup { "Low tempo" }
 c''4 d e f g1
 \tempoPadded #4.0 "High tempo"
 g4 f e d c1
}

Low tempo

High tempo
Anhang A  Notationsübersicht

A.1 Liste der Akkordbezeichnungen

Die Tabelle zeigt die zwei üblichen Möglichkeiten, wie Akkordbezeichnungen ausgegeben werden. Es wird auch die entsprechende Note ausgegeben.

<table>
<thead>
<tr>
<th>Ignatzek (default)</th>
<th>C</th>
<th>Cm</th>
<th>C+</th>
<th>C⁰</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative</td>
<td>C</td>
<td>C⁰3</td>
<td>C#5</td>
<td>C⁰5</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccccc}
\text{Def} & C^7 & Cm^7 & C^\Delta & C^9 & Cm^\Delta b5 \\
\text{Alt} & C^7 & C^7b3 & C^7#7 & C^7b3 b7 & C^7b3 b5 #7 \\
\end{array}
\]

<table>
<thead>
<tr>
<th>Ignatzek (default)</th>
<th>C</th>
<th>Cm</th>
<th>C+</th>
<th>C⁰</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative</td>
<td>C</td>
<td>C⁰6</td>
<td>C⁹</td>
<td>C⁹b3</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccccc}
\text{Def} & C^6 & Cm^6 & C^9 & Cm^9 \\
\text{Alt} & C^6 & C^6 b3 & C^9 & C^9 b3 \\
\end{array}
\]

<table>
<thead>
<tr>
<th>Ignatzek (default)</th>
<th>C</th>
<th>Cm</th>
<th>C+</th>
<th>C⁰</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative</td>
<td>C</td>
<td>C¹³</td>
<td>C¹¹</td>
<td>C¹³ b⁹</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccccc}
\text{Def} & C^¹³ b3 & C^¹¹ b3 & C^7 b⁵ b⁹ & C^7 b⁹ \\
\text{Alt} & C^¹³ b3 & C^¹¹ b3 & C^7 b⁵ b⁹ & C^7 b¹³ \\
\end{array}
\]

<table>
<thead>
<tr>
<th>Ignatzek (default)</th>
<th>C</th>
<th>Cm</th>
<th>C+</th>
<th>C⁰</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative</td>
<td>C</td>
<td>C⁷ #9</td>
<td>C¹¹</td>
<td>C⁷ #11</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccccc}
\text{Def} & C^7 #9 & C¹¹ & C^7 #11 & C¹³ \\
\text{Alt} & C^7 #9 & C¹¹ & C^7 #11 & C¹³ \\
\end{array}
\]

<table>
<thead>
<tr>
<th>Ignatzek (default)</th>
<th>C</th>
<th>Cm</th>
<th>C+</th>
<th>C⁰</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative</td>
<td>C</td>
<td>C⁷ #13</td>
<td>C⁷ #5 #9</td>
<td>C⁷ #13</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccccc}
\text{Def} & C^9 #13 b¹³ & C^7 #5 #9 & C^7 #9 #11 & C¹¹ b¹³ \\
\text{Alt} & C^9 #13 b¹³ & C^7 #5 #9 & C^7 #9 #11 & C¹¹ b¹³ \\
\end{array}
\]
### A.2 Übliche Akkord-Variablen

Die Tabelle zeigt Modifikatoren für Akkorde, die im `chordmode`-Modus benutzt werden können, um übliche Akkordkonstrukte zu notieren.

<table>
<thead>
<tr>
<th>Akkordtyp</th>
<th>Intervalle</th>
<th>Modifikator(en)</th>
<th>Beispiel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dur</td>
<td>große Terz, Quinte</td>
<td>5 oder nichts</td>
<td></td>
</tr>
<tr>
<td>Moll</td>
<td>kleine Terz, Quinte</td>
<td>m oder m5</td>
<td></td>
</tr>
</tbody>
</table>
Übernässig  
Große Terz, übermäßige aug 
Quinte 

Vermindert  
Kleine Terz, verminderte dim 
Quinte 

Dominantsieben  
Durdreiklang, kleine 7 
Septime 

Große Septime  
Durdreiklang, große maj7 oder maj 
Septime 

Kleine Septime  
Molldreiklang, kleine m7 
Septime 

Verminderte Septime  
Verminderter Dreiklang, dim7 
verminderte Septime 

Übernässige Septime  
Übernässiger Dreiklang, aug7 
kleine Septime 

halbverminderte Septime  
Verminderter Dreiklang, m7.5- 
kleine Sept 

Kleine MollSept  
Molldreiklang, große m7+ 
Septime 

Große Sexte  
Durdreiklang, Sexte 6
Kleine Sexte Molldreiklang, Sexte m6

Dominantnone Dominantsept, große 9 None

Dur-None Große None, große maj9 Septime

Moll-None Große None, kleine m9 Septime

Dominantundezime Dominantnone, perfekte 11 Undezime

Durundezime Große None, perfekte maj11 Undezime

Mollundezime Kleine None, perfekte m11 Undezime

Dominant-13 Dominantnone, große 13 13

Dominant-13 Dominant-Undezime, 13.11 große 13

Dur-13 Große Undezime, große 13 maj13.11
Moll-13 Kleine Undezime, große 13\textsuperscript{m}13.11

Sekundakkord große Sekunde, perfekte sus\textsuperscript{2}
Quinte

Quartakkord perfekte Quarte, perfekte sus\textsuperscript{4}
Quinte

Powerakkord Perfekte Quinte 1.5
(zweistimmig)

Powerakkord Perfekte Quinte, Oktave 1.5.8
(dreistimmig)

### A.3 Vordefinierte Saitenstimmungen

Die folgende Tabelle zeigt die vordefinierten Saitenstimmungen:

**Guitar tunings**

<table>
<thead>
<tr>
<th>guitar-tuning</th>
<th>guitar-seven-string-tuning</th>
<th>guitar-drop-d-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>guitar-drop-c-tuning</th>
<th>guitar-open-g-tuning</th>
<th>guitar-open-d-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>s</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>guitar-dadgad-tuning</th>
<th>guitar-lute-tuning</th>
<th>guitar-asus\textsuperscript{4}-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>s</td>
<td>s</td>
</tr>
</tbody>
</table>

**Bass tunings**

<table>
<thead>
<tr>
<th>bass-tuning</th>
<th>bass-four-string-tuning</th>
<th>bass-drop-d-tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>s</td>
<td></td>
</tr>
</tbody>
</table>
A.4 Die vordefinierten Bund-Diagramme

Die Tabelle zeigt alle vordefinierten Bunddiagramme für Gitarre.
Die folgende Tabelle zeigt vordefinierte Bunddiagramme für Ukulele.
Die folgende Tabelle zeigt die vordefinierten Bunddiagramme für Mandoline.
A.5 Vordefinierte Papierformate

Papierformate sind definiert in `scm/paper.scm`

Die „ISO 216“ A-Serie (DinA)

- "a10" (26 x 37 mm)
- "a9" (37 x 52 mm)
- "a8" (52 x 74 mm)
- "a7" (74 x 105 mm)
- "a6" (105 x 148 mm)
- "a5" (148 x 210 mm)
- "a4" (210 x 297 mm)
- "a3" (297 x 420 mm)
- "a2" (420 x 594 mm)
- "a1" (594 x 841 mm)
- "a0" (841 x 1189 mm)

Die „ISO 216“ B-Serie (DinB)

- "b10" (31 x 44 mm)
- "b9" (44 x 62 mm)
- "b8" (62 x 88 mm)
- "b7" (88 x 125 mm)
- "b6" (125 x 176 mm)
- "b5" (176 x 250 mm)
- "b4" (250 x 353 mm)
- "b3" (353 x 500 mm)
- "b2" (500 x 707 mm)
- "b1" (707 x 1000 mm)
- "b0" (1000 x 1414 mm)

Zwei erweiterte Größen wie definiert in „DIN 476“

- "4a0" (1682 x 2378 mm)
- "2a0" (1189 x 1682 mm)
### „ISO 269“ Standard-C-Serie (DinC)

<table>
<thead>
<tr>
<th>Format</th>
<th>Maße (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;c10&quot;</td>
<td>(28 x 40 mm)</td>
</tr>
<tr>
<td>&quot;c9&quot;</td>
<td>(40 x 57 mm)</td>
</tr>
<tr>
<td>&quot;c8&quot;</td>
<td>(57 x 81 mm)</td>
</tr>
<tr>
<td>&quot;c7&quot;</td>
<td>(81 x 114 mm)</td>
</tr>
<tr>
<td>&quot;c6&quot;</td>
<td>(114 x 162 mm)</td>
</tr>
<tr>
<td>&quot;c5&quot;</td>
<td>(162 x 229 mm)</td>
</tr>
<tr>
<td>&quot;c4&quot;</td>
<td>(229 x 324 mm)</td>
</tr>
<tr>
<td>&quot;c3&quot;</td>
<td>(324 x 458 mm)</td>
</tr>
<tr>
<td>&quot;c2&quot;</td>
<td>(458 x 648 mm)</td>
</tr>
<tr>
<td>&quot;c1&quot;</td>
<td>(648 x 917 mm)</td>
</tr>
<tr>
<td>&quot;c0&quot;</td>
<td>(917 x 1297 mm)</td>
</tr>
</tbody>
</table>

### Nordamerikanische Papierformate

<table>
<thead>
<tr>
<th>Format</th>
<th>Maße (in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;junior-legal&quot;</td>
<td>(8.0 x 5.0)</td>
</tr>
<tr>
<td>&quot;legal&quot;</td>
<td>(8.5 x 14.0)</td>
</tr>
<tr>
<td>&quot;ledger&quot;</td>
<td>(17.0 x 11.0)</td>
</tr>
<tr>
<td>&quot;letter&quot;</td>
<td>(8.5 x 11.0)</td>
</tr>
<tr>
<td>&quot;tabloid&quot;</td>
<td>(11.0 x 17.0)</td>
</tr>
<tr>
<td>&quot;11x17&quot;</td>
<td>(11.0 x 17.0)</td>
</tr>
<tr>
<td>&quot;17x11&quot;</td>
<td>(17.0 x 11.0)</td>
</tr>
</tbody>
</table>

### Government-letter der IEEE Printer Working Group, für Kinder zum Schreiben

<table>
<thead>
<tr>
<th>Format</th>
<th>Maße (in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;government-letter&quot;</td>
<td>(8 x 10.5)</td>
</tr>
<tr>
<td>&quot;government-legal&quot;</td>
<td>(8.5 x 13.0)</td>
</tr>
<tr>
<td>&quot;philippine-legal&quot;</td>
<td>(8.5 x 13.0)</td>
</tr>
</tbody>
</table>

### ANSI-Formate

<table>
<thead>
<tr>
<th>Format</th>
<th>Maße (in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;ansi a&quot;</td>
<td>(8.5 x 11.0)</td>
</tr>
<tr>
<td>&quot;ansi b&quot;</td>
<td>(17.0 x 11.0)</td>
</tr>
<tr>
<td>&quot;ansi c&quot;</td>
<td>(17.0 x 22.0)</td>
</tr>
<tr>
<td>&quot;ansi d&quot;</td>
<td>(22.0 x 34.0)</td>
</tr>
<tr>
<td>&quot;ansi e&quot;</td>
<td>(34.0 x 44.0)</td>
</tr>
<tr>
<td>&quot;engineering f&quot;</td>
<td>(28.0 x 40.0)</td>
</tr>
</tbody>
</table>
Nordamerikanische architektonische Formate

"arch a"  (9.0 x 12.0 in)
"arch b"  (12.0 x 18.0 in)
"arch c"  (18.0 x 24.0 in)
"arch d"  (24.0 x 36.0 in)
"arch e"  (36.0 x 48.0 in)
"arch e1" (30.0 x 42.0 in)

Alte Formate, die noch in Großbritannien benützt werden

"statement"  (5.5 x 8.5 in)
"half letter" (5.5 x 8.5 in)
"quarto"    (8.0 x 10.0 in)
"octavo"    (6.75 x 10.5 in)
"executive" (7.25 x 10.5 in)
"monarch"   (7.25 x 10.5 in)
"foolscap"  (8.27 x 13.0 in)
"folio"     (8.27 x 13.0 in)
"super-b"   (13.0 x 19.0 in)
"post"      (15.5 x 19.5 in)
"crown"     (15.0 x 20.0 in)
"large post"(16.5 x 21.0 in)
"demy"      (17.5 x 22.5 in)
"medium"    (18.0 x 23.0 in)
"broadsheet"(18.0 x 24.0 in)
"royal"     (20.0 x 25.0 in)
"elephant"  (23.0 x 28.0 in)
"double demy"(22.5 x 35.0 in)
"quad demy" (35.0 x 45.0 in)
"atlas"     (26.0 x 34.0 in)
"imperial" (22.0 x 30.0 in)

"antiquarian" (31.0 x 53.0 in)

**Auf PA4 basierende Formate**

"pa0" (840 x 1120 mm)
"pa1" (560 x 840 mm)
"pa2" (420 x 560 mm)
"pa3" (280 x 420 mm)
"pa4" (210 x 280 mm)
"pa5" (140 x 210 mm)
"pa6" (105 x 140 mm)
"pa7" (70 x 105 mm)
"pa8" (52 x 70 mm)
"pa9" (35 x 52 mm)
"pa10" (26 x 35 mm)

**In Südostasien und Australien benutzt**

"f4" (210 x 330 mm)

Benützt für sehr kleine @lilypond-Beispiele in der Dokumentation, basierend auf A8 quer

"a8landscape" (74 x 52 mm)

### A.6 MIDI-Instrumente

Hier eine Liste von Musikinstrumentenbezeichnungen, die als Name für `midiInstrument` benutzt werden können. Die Anordnung der Instrumente entspricht den 128 Programmnummern des MIDI-Standards.

<p>| acoustic grand | contrabass | lead 7 (fifths) |
| bright acoustic | tremolo strings | lead 8 (bass+lead) |
| electric grand | pizzicato strings | pad 1 (new age) |
| honky-tonk | orchestral harp | pad 2 (warm) |
| electric piano 1 | timpani | pad 3 (polysynth) |
| electric piano 2 | string ensemble 1 | pad 4 (choir) |
| harpsichord | string ensemble 2 | pad 5 (bowed) |
| clav | synthstrings 1 | pad 6 (metallic) |
| celesta | synthstrings 2 | pad 7 (halo) |
| glockenspiel | choir aahs | pad 8 (sweep) |
| music box | voice oohs | fx 1 (rain) |
| vibraphone | synth voice | 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) |</p>
<table>
<thead>
<tr>
<th>Instrument</th>
<th>Notation</th>
<th>Effect (description)</th>
</tr>
</thead>
<tbody>
<tr>
<td>percussive organ</td>
<td>french horn</td>
<td>fx 8 (sci-fi)</td>
</tr>
<tr>
<td>rock organ</td>
<td>brass section</td>
<td>sitar</td>
</tr>
<tr>
<td>church organ</td>
<td>synthbrass 1</td>
<td>banjo</td>
</tr>
<tr>
<td>reed organ</td>
<td>synthbrass 2</td>
<td>shanai</td>
</tr>
<tr>
<td>accordion</td>
<td>soprano sax</td>
<td>koto</td>
</tr>
<tr>
<td>harmonica</td>
<td>alto sax</td>
<td>kalimba</td>
</tr>
<tr>
<td>concertina</td>
<td>tenor sax</td>
<td>bagpipe</td>
</tr>
<tr>
<td>acoustic guitar (nylon)</td>
<td>baritone sax</td>
<td>fiddle</td>
</tr>
<tr>
<td>acoustic guitar (steel)</td>
<td>oboe</td>
<td>shanai</td>
</tr>
<tr>
<td>electric guitar (jazz)</td>
<td>english horn</td>
<td>tinkle bell</td>
</tr>
<tr>
<td>electric guitar (clean)</td>
<td>bassoon</td>
<td>agogo</td>
</tr>
<tr>
<td>electric guitar (muted)</td>
<td>clarinet</td>
<td>steel drums</td>
</tr>
<tr>
<td>overdriven guitar</td>
<td>piccolo</td>
<td>woodblock</td>
</tr>
<tr>
<td>distorted guitar</td>
<td>flute</td>
<td>taiko drum</td>
</tr>
<tr>
<td>guitar harmonics</td>
<td>recorder</td>
<td>melodic tom</td>
</tr>
<tr>
<td>acoustic bass</td>
<td>pan flute</td>
<td>synth drum</td>
</tr>
<tr>
<td>electric bass (finger)</td>
<td>blown bottle</td>
<td>reverse cymbal</td>
</tr>
<tr>
<td>electric bass (pick)</td>
<td>shakuhachi</td>
<td>guitar fret noise</td>
</tr>
<tr>
<td>fretless bass</td>
<td>whistle</td>
<td>breath noise</td>
</tr>
<tr>
<td>slap bass 1</td>
<td>ocarina</td>
<td>seashore</td>
</tr>
<tr>
<td>slap bass 2</td>
<td>lead 1 (square)</td>
<td>bird tweet</td>
</tr>
<tr>
<td>synth bass 1</td>
<td>lead 2 (sawtooth)</td>
<td>telephone ring</td>
</tr>
<tr>
<td>synth bass 2</td>
<td>lead 3 (calliope)</td>
<td>helicopter</td>
</tr>
<tr>
<td>violin</td>
<td>lead 4 (chiff)</td>
<td>applause</td>
</tr>
<tr>
<td>viola</td>
<td>lead 5 (charang)</td>
<td>gunshot</td>
</tr>
<tr>
<td>cello</td>
<td>lead 6 (voice)</td>
<td></td>
</tr>
</tbody>
</table>

### A.7 Liste der Farben

#### Normale Farben

Die Syntax zur Benutzung findet sich im Abschnitt [Farbige Objekte], Seite 208.

<table>
<thead>
<tr>
<th>Farbname</th>
<th>Erklärung</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>Das dunkelste Grau.</td>
</tr>
<tr>
<td>white</td>
<td>Das weißeste Grau.</td>
</tr>
<tr>
<td>red</td>
<td>Das rotste Grau.</td>
</tr>
<tr>
<td>green</td>
<td>Das grünest Grau.</td>
</tr>
<tr>
<td>blue</td>
<td>Das bluest Grau.</td>
</tr>
<tr>
<td>cyan</td>
<td>Blau mit Lila.</td>
</tr>
<tr>
<td>magenta</td>
<td>Lila mit Blau.</td>
</tr>
<tr>
<td>yellow</td>
<td>Gelb.</td>
</tr>
<tr>
<td>grey</td>
<td>Grau.</td>
</tr>
<tr>
<td>darkred</td>
<td>Dunkelrot.</td>
</tr>
<tr>
<td>darkgreen</td>
<td>Dunkelgrün.</td>
</tr>
<tr>
<td>darkblue</td>
<td>Dunkelblau.</td>
</tr>
<tr>
<td>darkmagenta</td>
<td>Dunkelrosa.</td>
</tr>
<tr>
<td>darkcyan</td>
<td>Dunkelcyan.</td>
</tr>
<tr>
<td>snow</td>
<td>Schnee.</td>
</tr>
<tr>
<td>OldLace</td>
<td>Alte Besensträhne.</td>
</tr>
<tr>
<td>OldLace</td>
<td>Linen.</td>
</tr>
<tr>
<td>bisque</td>
<td>Bisque.</td>
</tr>
<tr>
<td>ivory</td>
<td>Elfenbein.</td>
</tr>
<tr>
<td>azure</td>
<td>Blau.</td>
</tr>
<tr>
<td>FloralWhite</td>
<td>Blau mit Rosa.</td>
</tr>
<tr>
<td>BlanchedAlmond</td>
<td>Blaundunkelrosa.</td>
</tr>
<tr>
<td>cornsilk</td>
<td>Körner.</td>
</tr>
<tr>
<td>MintCream</td>
<td>Mintblau.</td>
</tr>
<tr>
<td>lavender</td>
<td>Lavendel.</td>
</tr>
<tr>
<td>LavenderBlush</td>
<td>Blasenlava.</td>
</tr>
</tbody>
</table>

#### 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“).
<table>
<thead>
<tr>
<th>Farben mit einer numeralen Endung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Für die folgenden Bezeichnungen kann das Suffix N durch eine Zahl von 1–4 ersetzt werden.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Farbe</th>
<th>Nachfolgende Bezeichnung</th>
</tr>
</thead>
<tbody>
<tr>
<td>snowN</td>
<td>seashellN</td>
</tr>
<tr>
<td>LavenderWhiteN</td>
<td>LemonChiffonN</td>
</tr>
<tr>
<td>LavenderBlushN</td>
<td>MistyRoseN</td>
</tr>
<tr>
<td>blueN</td>
<td>DodgerBlueN</td>
</tr>
<tr>
<td>LightSkyBlueN</td>
<td>LightSteelBlueN</td>
</tr>
<tr>
<td>CadetBlueN</td>
<td>turquoiseN</td>
</tr>
<tr>
<td>SeaGreenN</td>
<td>PaleGreenN</td>
</tr>
<tr>
<td>OliveDrabN</td>
<td>DarkOliveGreenN</td>
</tr>
<tr>
<td>yellowN</td>
<td>goldN</td>
</tr>
<tr>
<td>IndianRedN</td>
<td>siennaN</td>
</tr>
<tr>
<td>chocolateN</td>
<td>firebrickN</td>
</tr>
<tr>
<td>orangeN</td>
<td>DarkOrangeN</td>
</tr>
<tr>
<td>redN</td>
<td>DeepPinkN</td>
</tr>
<tr>
<td>PaleVioletRedN</td>
<td>maroonN</td>
</tr>
<tr>
<td>plumN</td>
<td>MediumOrchidN</td>
</tr>
<tr>
<td>thistleN</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grauskala</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eine Grauskala kann mit der Bezeichnung</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Farbe</th>
<th>Bezeichnung</th>
</tr>
</thead>
<tbody>
<tr>
<td>greyN</td>
<td></td>
</tr>
</tbody>
</table>

erstellt werden, wobei N eine Zahl von 0–100 darstellt.
A.8 Die Emmentaler-Schriftart

Die folgenden Symbole sind als Emmentaler-Schriftart verfügbar; auf sie kann direkt zugegriffen werden, indem man die übliche Textbeschriftung benutzt. \texttt{musicglyph} greift direkt auf die Notationsschriftart zu

\texttt{g^\markup{\texttt{musicglyph} "scripts.segno" }}

oder

\texttt{\markup{\texttt{musicglyph} "five"}}.

Siehe auch Abschnitt 1.8.2 [Text formatieren], Seite 223.

Notenschlüssel-Glyphen

\begin{itemize}
  \item \texttt{clefs.C}
  \item \texttt{clefs.varC}
  \item \texttt{clefs.F}
  \item \texttt{clefs.G}
  \item \texttt{clefs.GG}
  \item \texttt{clefs.tenorG}
  \item \texttt{clefs.percussion}
  \item \texttt{clefs.varpercussion}
  \item \texttt{clefs.tab}
\end{itemize}

Taktart-Glyphen

\begin{itemize}
  \item \texttt{timesig.C44}
  \item \texttt{timesig.C22}
\end{itemize}
Zahlen-Glyphen

plus   +  comma   ,

hyphen  -  period  .

zero  0  one  1

two  2  three  3

four  4  five  5

six  6  seven  7

eight  8  nine  9

Versetzungszeichen-Glyphen

accidentals.sharp  #  accidentals
 .sharp.arrowup  

accidentals .sharp.arrowdown  #  accidentals
 .sharp.arrowboth  

accidentals.sharp .slashslash.stem  #  accidentals.sharp
 .slashslashslash.stemstem  

accidentals.sharp .slashslashslash.stem  #  accidentals.sharp
 .slashslashlash.stemstemstem  

accidentals.doublesharp  ✷  accidentals.natural  ♮

accidentals .natural.arrowup  ♩  accidentals
 .natural.arrowdown  ♩

accidentals .natural.arrowboth  ↓  accidentals.flat  ♩
### Accidentals

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accidentals.flat.arrowup</td>
<td>accidentals.flat.arrowdown</td>
</tr>
<tr>
<td>accidentals.flat.arrowboth</td>
<td>accidentals.flat.slash</td>
</tr>
<tr>
<td>accidentals.flat.slashslash</td>
<td>accidentals.mirroredflat.flat</td>
</tr>
<tr>
<td>accidentals.mirroredflat.backslash</td>
<td>accidentals.flatflat</td>
</tr>
<tr>
<td>accidentals.flatflat.slash</td>
<td>accidentals.rightparen</td>
</tr>
<tr>
<td>accidentals.leftparen</td>
<td></td>
</tr>
</tbody>
</table>

### Standard-Notenkopf-Glyphen

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>noteheads.uM2</td>
<td>noteheads.dm2</td>
</tr>
<tr>
<td>noteheads.sM1</td>
<td>noteheads.s0</td>
</tr>
<tr>
<td>noteheads.s1</td>
<td>noteheads.s2</td>
</tr>
</tbody>
</table>

### Spezielle Notenkopf-Glyphen

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>noteheads.sM1double</td>
<td>noteheads.s0diamond</td>
</tr>
<tr>
<td>noteheads.s1diamond</td>
<td>noteheads.s2diamond</td>
</tr>
<tr>
<td>noteheads.s0triangle</td>
<td>noteheads.dltriangle</td>
</tr>
<tr>
<td>noteheads.ultriangle</td>
<td>noteheads.u2triangle</td>
</tr>
<tr>
<td>noteheads.d2triangle</td>
<td>noteheads.s0slash</td>
</tr>
</tbody>
</table>
### Anhang A: Notationsbersicht

<table>
<thead>
<tr>
<th>Notation 1</th>
<th>Notation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>noteheads.s1slash</code></td>
<td><code>noteheads.s2slash</code></td>
</tr>
<tr>
<td><code>noteheads.s0cross</code></td>
<td><code>noteheads.s1cross</code></td>
</tr>
<tr>
<td><code>noteheads.s2cross</code></td>
<td><code>noteheads.s2xcircle</code></td>
</tr>
<tr>
<td><code>noteheads.s0harmonic</code></td>
<td><code>noteheads.s2harmonic</code></td>
</tr>
</tbody>
</table>

### Geformte Notenkopf-Glyphen

<table>
<thead>
<tr>
<th>Notation 1</th>
<th>Notation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>noteheads.s0do</code></td>
<td><code>noteheads.d1do</code></td>
</tr>
<tr>
<td><code>noteheads.u1do</code></td>
<td><code>noteheads.d2do</code></td>
</tr>
<tr>
<td><code>noteheads.u2do</code></td>
<td><code>noteheads.s0doThin</code></td>
</tr>
<tr>
<td><code>noteheads.d1doThin</code></td>
<td><code>noteheads.u1doThin</code></td>
</tr>
<tr>
<td><code>noteheads.d2doThin</code></td>
<td><code>noteheads.u2doThin</code></td>
</tr>
<tr>
<td><code>noteheads.s0re</code></td>
<td><code>noteheads.ulre</code></td>
</tr>
<tr>
<td><code>noteheads.d1re</code></td>
<td><code>noteheads.u2re</code></td>
</tr>
<tr>
<td><code>noteheads.d2re</code></td>
<td><code>noteheads.s0reThin</code></td>
</tr>
<tr>
<td><code>noteheads.ulreThin</code></td>
<td><code>noteheads.d1reThin</code></td>
</tr>
<tr>
<td><code>noteheads.u2reThin</code></td>
<td><code>noteheads.d2reThin</code></td>
</tr>
<tr>
<td><code>noteheads.s0mi</code></td>
<td><code>noteheads.s1mi</code></td>
</tr>
<tr>
<td>notation</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-----</td>
</tr>
<tr>
<td>noteheads.s2mi</td>
<td></td>
</tr>
<tr>
<td>noteheads.s1miMirror</td>
<td></td>
</tr>
<tr>
<td>noteheads.s0miThin</td>
<td></td>
</tr>
<tr>
<td>noteheads.s2miThin</td>
<td></td>
</tr>
<tr>
<td>noteheads.d0fa</td>
<td></td>
</tr>
<tr>
<td>noteheads.d1fa</td>
<td></td>
</tr>
<tr>
<td>noteheads.d2fa</td>
<td></td>
</tr>
<tr>
<td>noteheads.d0faThin</td>
<td></td>
</tr>
<tr>
<td>noteheads.d1faThin</td>
<td></td>
</tr>
<tr>
<td>noteheads.d2faThin</td>
<td></td>
</tr>
<tr>
<td>noteheads.s0sol</td>
<td></td>
</tr>
<tr>
<td>noteheads.s1sol</td>
<td></td>
</tr>
<tr>
<td>noteheads.s0la</td>
<td></td>
</tr>
<tr>
<td>noteheads.s2la</td>
<td></td>
</tr>
<tr>
<td>noteheads.s1laThin</td>
<td></td>
</tr>
<tr>
<td>noteheads.s0ti</td>
<td></td>
</tr>
<tr>
<td>noteheads.dlti</td>
<td>noteheads.u2ti</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>noteheads.d2ti</td>
<td>noteheads.s0tiThin</td>
</tr>
<tr>
<td>noteheads.ultiThin</td>
<td>noteheads.dltiThin</td>
</tr>
<tr>
<td>noteheads.u2tiThin</td>
<td>noteheads.d2tiThin</td>
</tr>
<tr>
<td>noteheads.u0doFunk</td>
<td>noteheads.d0doFunk</td>
</tr>
<tr>
<td>noteheads.u1doFunk</td>
<td>noteheads.d1doFunk</td>
</tr>
<tr>
<td>noteheads.u2doFunk</td>
<td>noteheads.d2doFunk</td>
</tr>
<tr>
<td>noteheads.u0reFunk</td>
<td>noteheads.d0reFunk</td>
</tr>
<tr>
<td>noteheads.u1reFunk</td>
<td>noteheads.d1reFunk</td>
</tr>
<tr>
<td>noteheads.u2reFunk</td>
<td>noteheads.d2reFunk</td>
</tr>
<tr>
<td>noteheads.u0miFunk</td>
<td>noteheads.d0miFunk</td>
</tr>
<tr>
<td>noteheads.ulmiFunk</td>
<td>noteheads.d1miFunk</td>
</tr>
<tr>
<td>noteheads.s2miFunk</td>
<td>noteheads.u0faFunk</td>
</tr>
<tr>
<td>noteheads.d0faFunk</td>
<td>noteheads.ulfaFunk</td>
</tr>
<tr>
<td>noteheads.d1faFunk</td>
<td>noteheads.u2faFunk</td>
</tr>
</tbody>
</table>
Pausen-Glyphen

```
rests.0  -  rests.1
rests.0o -  rests.1o
rests.M3 II  rests.M2
rests.M1  ■  rests.M1o
rests.2  \  rests.2classical
rests.2z  \  rests.3
rests.4  \  rests.5
rests.6  \  rests.7
```
Fähnchen-Glyphen

flags.u3 /flags.u4
flags.u5 /flags.u6
flags.u7 /flags.d3
flags.d4 /flags.d5
flags.d6 /flags.d7
flags.ugrace /flags.dgrace

Punkt-Glyphen

dots.dot

Dynamik-Glyphen

space \ f \ f
m \ m \ p \ p
r \ r \ s \ s
z \ z
Anhang A: Notationsbersicht

Schrift-Glyphen

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.uportato  ◡
scripts.dportato  ◡  scripts.umarcato  ◡
scripts.dmarcato  ◡  scripts.open  ◡
scripts.halfopen  ◡  scripts.halfopenvertical  ◡
scripts.stopped  ◡  scripts.upbow  ◡
scripts.downbow  ◡  scripts.reverseturn  ◡
scripts.turn  ◡  scripts.trill  ◡
scripts.caesura.straight // scripts.tickmark ✓

scripts.snappizzicato ◆ scripts.ictus ∘

scripts.uaccentus ◆ scripts.daccentus ∘

scripts.usemicirculus ◆ scripts.dsemicirculus ∘

scripts.circulus ◆ scripts.augmentum ∘

scripts.usignumcongruentiae ◆ scripts.dsignumcongruentiae §

Pfeilkopf-Glyphen

arrowheads.open.01 ◄ arrowheads.open.0M1 ◄

arrowheads.open.11 ◄ arrowheads.open.1M1 ◄

arrowheads.close.01 ◄ arrowheads.close.0M1 ◄

arrowheads.close.11 ◄ arrowheads.close.1M1 ◄

Klammerspitzen-Glyphen

brackettips.up ◄ brackettips.down ◄

Pedal-Glyphen

pedal.* ◆ pedal.M ∗

pedal.. ◆ pedal.P ∗

pedal.d ◆ pedal.e ∗

pedal.Ped ◆
Akkordeon-Glyphen

- accordion.discant
- accordion.freebass
- accordion.bayanbass
- accordion.push

Bogen-Glyphen

- ties.lyric.short
- ties.lyric.default

Vaticana-Glyphen

- clefs.vaticana.do
- clefs.vaticana.fa
- custodes.vaticana.u0
- custodes.vaticana.u2
- custodes.vaticana.d1
- accidentals.vaticanaM1
- dots.dotvaticana
- noteheads.svaticana.punctum
- noteheads.svaticana.punctum.cavum
- noteheads.svaticana.linea.punctum
Anhang A: Notationsbersicht

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.epiphonus
noteheads
  .svaticana.reverse.plica
noteheads
  .svaticana.inner.cephalicus

noteheads
  .svaticana.quilisma

Medicaea-Glyphen

clefs.medicaea.do
  clefs.medicaea.do_change

clefs.medicaea.fa
  clefs.medicaea.fa_change

custodes.medicaea.u0
  custodes.medicaea.u1

custodes.medicaea.u2
  custodes.medicaea.d0

custodes.medicaea.d1
  custodes.medicaea.d2

accidentals.medicaeaM1
  noteheads.smedicaea.
    .inclinatum

noteheads
  .smedicaea.punctum
noteheads
  .smedicaea.rvirga

noteheads
  .smedicaea.virga
### Hufnagel-Glyphen

<table>
<thead>
<tr>
<th>clefs.hufnagel.do</th>
<th>clefs.hufnagel.do_change</th>
</tr>
</thead>
<tbody>
<tr>
<td>clefs.hufnagel.fa</td>
<td>clefs.hufnagel.fa_change</td>
</tr>
<tr>
<td>clefs.hufnagel.do.fa</td>
<td>clefs.hufnagel.do.fa_change</td>
</tr>
<tr>
<td>custodes.hufnagel.u0</td>
<td>custodes.hufnagel.u1</td>
</tr>
<tr>
<td>custodes.hufnagel.u2</td>
<td>custodes.hufnagel.d0</td>
</tr>
<tr>
<td>custodes.hufnagel.d1</td>
<td>custodes.hufnagel.d2</td>
</tr>
<tr>
<td>accidentals.hufnagelM1</td>
<td>noteheads.shufnagel.punctum</td>
</tr>
<tr>
<td>noteheads.shufnagel.virga</td>
<td>noteheads.shufnagel.lpes</td>
</tr>
</tbody>
</table>

### Mensural-Glyphen

<table>
<thead>
<tr>
<th>rests.M3mensural</th>
<th>rests.M2mensural</th>
</tr>
</thead>
<tbody>
<tr>
<td>rests.M1mensural</td>
<td>rests.0mensural</td>
</tr>
<tr>
<td>rests.1mensural</td>
<td>rests.2mensural</td>
</tr>
<tr>
<td>rests.3mensural</td>
<td>rests.4mensural</td>
</tr>
<tr>
<td>clefs.mensural.c</td>
<td>clefs.mensural.c_change</td>
</tr>
<tr>
<td>clefs.blackmensural.c</td>
<td>clefs.blackmensural.c_change</td>
</tr>
</tbody>
</table>
Anhang A: Notationsbersicht

<table>
<thead>
<tr>
<th>clefs.mensural.f</th>
<th>clefs.mensural.f_change</th>
</tr>
</thead>
<tbody>
<tr>
<td>clefs.mensural.g</td>
<td>clefs.mensural.g_change</td>
</tr>
<tr>
<td>custodes.mensural.u0</td>
<td>custodes.mensural.ul</td>
</tr>
<tr>
<td>custodes.mensural.u2</td>
<td>custodes.mensural.d0</td>
</tr>
<tr>
<td>custodes.mensural.d1</td>
<td>custodes.mensural.d2</td>
</tr>
<tr>
<td>accidentals.mensuralu1</td>
<td>accidentals.mensuralM1</td>
</tr>
<tr>
<td>flags.mensuralu03</td>
<td>flags.mensuralu13</td>
</tr>
<tr>
<td>flags.mensuralu23</td>
<td>flags.mensurald03</td>
</tr>
<tr>
<td>flags.mensurald13</td>
<td>flags.mensurald23</td>
</tr>
<tr>
<td>flags.mensuralu04</td>
<td>flags.mensuralu14</td>
</tr>
<tr>
<td>flags.mensuralu24</td>
<td>flags.mensurald04</td>
</tr>
<tr>
<td>flags.mensurald14</td>
<td>flags.mensurald24</td>
</tr>
<tr>
<td>flags.mensuralu05</td>
<td>flags.mensuralu15</td>
</tr>
<tr>
<td>flags.mensuralu25</td>
<td>flags.mensurald05</td>
</tr>
<tr>
<td>flags.mensurald15</td>
<td>flags.mensurald25</td>
</tr>
<tr>
<td>Anhang A: Notationsbersicht</td>
<td>641</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>flags.mensuralu06</td>
<td>flags.mensuralu16</td>
</tr>
<tr>
<td>flags.mensuralu26</td>
<td>flags.mensurald06</td>
</tr>
<tr>
<td>flags.mensurald16</td>
<td>flags.mensurald26</td>
</tr>
<tr>
<td>timesig.mensural44</td>
<td>timesig.mensural22</td>
</tr>
<tr>
<td>timesig.mensural32</td>
<td>timesig.mensural64</td>
</tr>
<tr>
<td>timesig.mensural94</td>
<td>timesig.mensural134</td>
</tr>
<tr>
<td>timesig.mensural68</td>
<td>timesig.mensural98</td>
</tr>
<tr>
<td>timesig.mensural48</td>
<td>timesig.mensural68alt</td>
</tr>
<tr>
<td>timesig.mensural24</td>
<td>noteheads.uM3mensural</td>
</tr>
<tr>
<td>noteheads.dM3mensural</td>
<td>noteheads.sM3ligmensural</td>
</tr>
<tr>
<td>noteheads.uM2mensural</td>
<td>noteheads.dM2mensural</td>
</tr>
<tr>
<td>noteheads.sM2ligmensural</td>
<td>noteheads.sM1mensural</td>
</tr>
<tr>
<td>noteheads.urM3mensural</td>
<td>noteheads.drM3mensural</td>
</tr>
<tr>
<td>noteheads.srM3ligmensural</td>
<td>noteheads.urM2mensural</td>
</tr>
<tr>
<td>noteheads.drM2mensural</td>
<td>noteheads.srM2ligmensural</td>
</tr>
<tr>
<td>noteheads.srM1mensural</td>
<td>noteheads.uM3semimensural</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>noteheads.dM3semimensural</td>
<td>noteheads.sM3semiligmensural</td>
</tr>
<tr>
<td>noteheads.uM2semimensural</td>
<td>noteheads.dM2semimensural</td>
</tr>
<tr>
<td>noteheads.sM2semiligmensural</td>
<td>noteheads.sM1semimensural</td>
</tr>
<tr>
<td>noteheads.urM3semimensural</td>
<td>noteheads.drM3semimensural</td>
</tr>
<tr>
<td>noteheads.srM3semiligmensural</td>
<td>noteheads.urM2semimensural</td>
</tr>
<tr>
<td>noteheads.drM2semimensural</td>
<td>noteheads.srM2semiligmensural</td>
</tr>
<tr>
<td>noteheads.srM1semimensural</td>
<td>noteheads.uM3blackmensural</td>
</tr>
<tr>
<td>noteheads.dM3blackmensural</td>
<td>noteheads.sM3blackligmensural</td>
</tr>
<tr>
<td>noteheads.uM2blackmensural</td>
<td>noteheads.dM2blackmensural</td>
</tr>
<tr>
<td>noteheads.sM2blackligmensural</td>
<td>noteheads.sM1blackmensural</td>
</tr>
<tr>
<td>noteheads.s0mensural</td>
<td>noteheads.s1mensural</td>
</tr>
<tr>
<td>noteheads.s2mensural</td>
<td>noteheads.s0blackmensural</td>
</tr>
</tbody>
</table>
Neomensural-Glyphen

rests.M3neomensural \(\text{II}\) rests.M2neomensural \(\text{I}\)

rests.M1neomensural \(\text{II}\) rests.0neomensural \(\text{I}\)

rests.1neomensural \(\text{I}\) rests.2neomensural \(\text{I}\)

rests.3neomensural \(\text{I}\) rests.4neomensural \(\text{I}\)

clefs.neomensural.c \(\text{V}\) clefs.neomensural.c_change \(\text{V}\)

timesig.neomensural44 \(\text{C}\) timesig.neomensural22 \(\text{C}\)

timesig.neomensural32 \(\text{O}\) timesig.neomensural64 \(\text{C}\)

timesig.neomensural94 \(\text{O}\) timesig.neomensural34 \(\text{O}\)

timesig.neomensural68 \(\text{C}\) timesig.neomensural98 \(\text{O}\)

timesig.neomensural48 \(\text{C}\) timesig.neomensural68alt \(\text{O}\)

timesig.neomensural24 \(\text{O}\) noteheads.uM3neomensural \(\text{I}\)

noteheads.dM3neomensural \(\text{II}\) noteheads.uM2neomensural \(\text{II}\)

noteheads.dM2neomensural \(\text{II}\) noteheads.sM1neomensural \(\text{II}\)

noteheads.urM3neomensural \(\text{II}\) noteheads.drM3neomensural \(\text{II}\)
Anhang A: Notationsbersicht

### Noteheads

- urM2neomensural
- drM2neomensural
- srM1neomensural
- s0neomensural
- s1neomensural
- s2neomensural

### Petrucci-Glyphen

- clefs.petrucci.c1
- clefs.petrucci.c1_change
- clefs.petrucci.c2
- clefs.petrucci.c2_change
- clefs.petrucci.c3
- clefs.petrucci.c3_change
- clefs.petrucci.c4
- clefs.petrucci.c4_change
- clefs.petrucci.c5
- clefs.petrucci.c5_change
- clefs.petrucci.f
- clefs.petrucci.f_change
- clefs.petrucci.g
- clefs.petrucci.g_change

- noteheads.s0petrucci
- noteheads.s1petrucci
- noteheads.s2petrucci

- noteheads.s0blackpetrucci
- noteheads.s1blackpetrucci
- noteheads.s2blackpetrucci
Solesmes-Glyphen

- noteheads.ssolesmes.incl.parvum
- noteheads.ssolesmes.auct.asc
- noteheads.ssolesmes.auct.desc
- noteheads.ssolesmes.stropha
- noteheads.ssolesmes.aucta
- noteheads.ssolesmes.oriscus

Glyphen der Kiever Notation

- clefs.kievan.do
- clefs.kievan.do_change
- accidentals.kievan1
- accidentals.kievanM1
- scripts.barline.kievan
- dots.dotkievan
- noteheads.sM2kievan
- noteheads.sM1kievan
- noteheads.s0kievan
- noteheads.d2kievan
- noteheads.u2kievan
- noteheads.s1kievan
- noteheads.srlkievan
- noteheads.d3kievan
- noteheads.u3kievan
A.9 Notenkopfstile

Folgende Stile können zur Darstellung der Notenköpfe verwendet werden:

- default
- altdefault
- baroque
- neomensural
- mensural
- petrucci
- harmonic
- harmonic-black
- harmonic-mixed
- diamond
- cross
- xcircle
- triangle
- slash

A.10 Textbeschriftungsbefehle

The following commands can all be used inside \markup { }. 

A.10.1 Font

\abs-fontsize size (number) arg (markup)

Use size as the absolute font size (in points) to display arg. Adjusts baseline-skip and word-space accordingly.

\markup {
  default text font size
  \hspace #2
}
Anhang A: Notationsübersicht

\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

Used properties:
• baseline-skip (3)
• word-space (0.6)
\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 {
  \caps {
    Text in small caps
  }
}

default  TEXT IN SMALL CAPS
\texttt{\textbackslash dynamic arg (markup)}

Use the dynamic font. This font only contains \texttt{s, f, m, z, p,} and \texttt{r}. When producing phrases, like ,\texttt{pi u f}, the normal words (like ,\texttt{pi u'}) should be done in a different font. The recommended font for this is bold and italic.

\texttt{\textbackslash markup \{ \textbackslash dynamic \{ \texttt{sfzp} \} \}}

\texttt{sfzp}

\texttt{\texttt{\textbackslash finger arg (markup)}}

Set \texttt{arg} as small numbers.

\texttt{\textbackslash markup \{ \textbackslash finger \{ \texttt{1 2 3 4 5} \} \}}

\texttt{1 2 3 4 5}

\texttt{\texttt{\textbackslash fontCaps arg (markup)}}

Set font\texttt{-shape} to caps

Note: \texttt{\textbackslash fontCaps} requires the installation and selection of fonts which support the \texttt{cap} font shape.

\texttt{\texttt{\textbackslash fontsize increment (number) arg (markup)}}

Add \texttt{increment} to the font\texttt{-size}. Adjusts baseline-skip accordingly.

\texttt{\textbackslash markup \{ \texttt{default \textbackslash hspace \#2 \textbackslash fontsize \#-1.5 smaller} } \texttt{default smaller} \texttt{\texttt{}}} \texttt{\texttt{}}

Used properties:

- \texttt{baseline-skip \(2\)}
- \texttt{word-space \(1\)}
- \texttt{font-size \(0\)}

\texttt{\texttt{\textbackslash huge arg (markup)}}

Set font size to \(+2\).

\texttt{\textbackslash markup \{ \texttt{default \textbackslash hspace \#2 \textbackslash huge huge} } \texttt{\texttt{}}} \texttt{\texttt{}}}
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)
Increase the font size relative to the current setting.
\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
  }
}

\textbf{some bold text} medium font series \textbf{bold again}

\normal-size-sub arg (markup)

Set arg in subscript with a normal font size.
\markup {
  \default {
    \normal-size-sub {
      subscript in standard size
    }
  }
}

default subscript in standard size

Used properties:
• font-size (0)

\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:
• font-size (0)

\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 {
    huge bold sans caps
    \hspace #2
    \normal-text {

huge normal
}\hspace #2
as before
}\}

\section*{HUGE BOLD SANS CAPS  huge normal  AS BEFORE}

\texttt{\normalsize arg} (markup)
Set font size to default.
\begin{verbatim}
\markup {
  \teeny {
    this is very small
  \hspace #2
  \normalsize {
    normal size
  }
  \hspace #2
  teeny again
  }
}
\end{verbatim}
this is very small  \texttt{normal size}  teeny again

\texttt{\number arg} (markup)
Set font family to \texttt{number}, which yields the font used for time signatures and fingerings. This font contains numbers and some punctuation; it has no letters.
\begin{verbatim}
\markup {
  \number {
    0 1 2 3 4 5 6 7 8 9 . ,
  }
}
\end{verbatim}
0123456789.,

\texttt{\overtie arg} (markup)
Overtie arg.
\begin{verbatim}
\markup \line {
  \overtie "overtied"
  \override #'(offset . 5)
  \override #'(thickness . 1)
  \overtie "overtied"
  \override #'(offset . 1)
  \override #'(thickness . 5)
  \overtie "overtied"
}
\end{verbatim}
overtied overtied overtied

Used properties:
\begin{itemize}
  \item shorten-pair (0 0)
\end{itemize}
• direction (1)
• offset (2)
• thickness (1)

\replace replacements (list) arg (markup)
Used to automatically replace a string by another in the markup arg. Each pair of
the alist replacements specifies what should be replaced. The key is the string to be
replaced by the value string.

\markup { \replace #'("thx" . "Thanks!") } thx

Thanks!

\roman arg (markup)
Set font family to roman.

\markup { \roman { text in roman font family } return to sans }

sans serif, bold text in roman font family return to sans

\sans arg (markup)
Switch to the sans serif font family.

\markup { \sans { sans serif 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
\texttt{\textbackslash small arg (markup)}

Set font size to -1.

\begin{verbatim}
\texttt{\textbackslash markup \{\texttt{default}\texttt{hspace #2}\texttt{\small small}\}}
\end{verbatim}

\texttt{default \small}

\texttt{\textbackslash smallCaps arg (markup)}

Emit \texttt{arg} as small caps.

Note: \texttt{\smallCaps} does not support accented characters.

\begin{verbatim}
\texttt{\textbackslash markup \{\texttt{default}\texttt{hspace #2}\texttt{\smallCaps \texttt{Text in small caps}}\}}
\end{verbatim}

\texttt{default TEXT IN SMALL CAPS}

\texttt{\textbackslash smaller arg (markup)}

Decrease the font size relative to the current setting.

\begin{verbatim}
\texttt{\textbackslash markup \{\texttt{\fontsize #3.5 \{\texttt{some large text}\texttt{\hspace #2}\texttt{\smaller \{\texttt{a bit smaller}}\texttt{\}}\texttt{\hspace #2}\texttt{more large text}}\}}
\end{verbatim}

\texttt{some large text a bit smaller more large text}

\texttt{\textbackslash sub arg (markup)}

Set \texttt{arg} in subscript.

\begin{verbatim}
\texttt{\textbackslash markup \{\texttt{\concat \texttt{H\texttt{\sub \{\texttt{2}\texttt{\}}\texttt{\}}\texttt{\}}\}}
\end{verbatim}
\begin{document}

$\text{H}_2\text{O}$

Used properties:
- \texttt{font-size (0)}

\texttt{\super arg (markup)}
Set \texttt{arg} in superscript.
\begin{verbatim}
\markup {
E = 
concat {
mc 
\super 2 
}
}
\end{verbatim}

$E = mc^2$

Used properties:
- \texttt{font-size (0)}

\texttt{\teeny arg (markup)}
Set font size to -3.
\begin{verbatim}
\markup {
  \default 
  \hspace #2
  \teeny 
  teeny
}
\end{verbatim}

\texttt{default teeny}

\texttt{\text arg (markup)}
Use a text font instead of music symbol or music alphabet font.
\begin{verbatim}
\markup {
  \number {
    1, 2, 
    \text {
      three, four, 
    }
    5
  }
}
\end{verbatim}

\texttt{1,2, three, four, 5}

\texttt{\tie arg (markup)}
Adds a horizontal bow created with \texttt{make-tie-stencil} at bottom or top of \texttt{arg}. Looks at \texttt{thickness} to determine line thickness, and \texttt{offset} to determine y-offset. The added bow fits the extent of \texttt{arg}, \texttt{shorten-pair} may be used to modify this. \texttt{direction} may be set using an \texttt{override} or direction-modifiers or \texttt{voiceOne}, etc.
\begin{verbatim}
\markup {
}
\end{verbatim}

\end{document}
\override #'(direction . 1) 
\tie "above"
\override #'(direction . -1) 
\tie "below"

\above \below

Used properties:
- shorten-pair ((0 . 0))
- direction (1)
- offset (2)
- thickness (1)

\tiny \arg (markup)
Set font size to -2.
\markup {
  \default
  \hspace #2
  \tiny
  \arg
}

\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, offset to determine line y-offset from \arg and underline-skip to determine the distance of additional lines from the others. underline-shift is used to get subsequent calls correct. Overriding it makes little sense, it would end up adding the provided value to the one of offset.
\markup \fill-line {
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(thickness . 1)
  \underline "underlined"
  \override #'(offset . 1)
  \override #'(thickness . 5)
  \underline "underlined"
  \override #'(offset . 5)
\override #'(underline-skip . 4)
\underline \underline \underline "multiple underlined"

\underline arg (markup)
\markup \line {
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(thickness . 1)
  \underline "underlined"
  \override #'(offset . 1)
  \override #'(thickness . 5)
  \underline "underlined"
}

\underline arg (markup)
\underline \underline \underline "multiple underlined"

\undertie arg (markup)
\markup \line {
  \undertie "undertied"
  \override #'(offset . 5)
  \override #'(thickness . 1)
  \undertie "undertied"
  \override #'(offset . 1)
  \override #'(thickness . 5)
  \undertie "undertied"
}

\undertie arg (markup)
\undertie \undertie \undertie "undertied"

\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
  }
}

\upright arg (markup)
\upright \upright \upright "upright"

A.10.2 Align
\center-align arg (markup)
Align arg to its X center.
\markup {

\column {  
one  
\center-align  
two  
\center-align  
three  
}  
}  
one  
two  
three  
  
\center-column { markup list}  
Put \texttt{args} in a centered column.  
\markup {  
\center-column {  
one  
two  
three  
}  
}  
one  
two  
three  
  
\texttt{args} (markup list)  
Used properties:  
• \texttt{baseline-skip}  

\column { markup list}  
Stack the markups in \texttt{args} vertically. The property \texttt{baseline-skip} determines the space between markups in \texttt{args}.  
\markup {  
\column {  
one  
two  
three  
}  
}  
one  
two  
three  
  
\texttt{columns} (markup list)  
Used properties:  
• \texttt{baseline-skip}  

\combine arg1 (markup) arg2 (markup)  
Print two markups on top of each other.  
Note: \texttt{\combine} cannot take a list of markups enclosed in curly braces as an argument; for this purpose use \texttt{\overlay} instead.  
\markup {
\concat args (markup list)

Concatenate args in a horizontal line, without spaces in between. 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 (markup list)

Make a column of args, going up or down, depending on the setting of the direction layout property.

\markup {
  \override #(direction . ,UP) {
    \dir-column {
      going up
    }
  }
  \hspace #1
  \dir-column {
    going down
  }
  \hspace #1
  \override #(direction . 1) {
    \dir-column {
      going up
    }
  }
}

up       up
going going going
down

Used properties:

- baseline-skip
- direction
\fill-line args (markup list)

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 (0.6)
- text-direction (1)

\fill-with-pattern space (number) dir (direction) pattern (markup) left (markup) right (markup)

Put left and right in a horizontal line of width line-width with a line of markups pattern in between. Patterns are spaced apart by space. Patterns are aligned to the dir markup.

\markup \column {
  "right-aligned :"
  \fill-with-pattern #1 #RIGHT . first right
  \fill-with-pattern #1 #RIGHT . second right
  \null
  "center-aligned :"
  \fill-with-pattern #1.5 #CENTER - left right
  \null
  "left-aligned :"
  \override #'(line-width . 50)
  \fill-with-pattern #2 #LEFT : left first
  \override #'(line-width . 50)
  \fill-with-pattern #2 #LEFT : left second
}

right-aligned:
first ......................................................... right
second ......................................................... right

center-aligned:
left - - - - - - - - - - - - - - - - - - - - - - - right

left-aligned:

Used properties:

- line-width
- word-space

\general-align \textit{axis} (integer) \textit{dir} (number) \textit{arg} (markup)
Align \textit{arg} in \textit{axis} direction to the \textit{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
  }
}
}
\texttt{\textbackslash{halign} \textit{dir} (number) \textit{arg} (markup)}

Set horizontal alignment. If \textit{dir} is -1, then it is left-aligned, while +1 is right. Values in between interpolate alignment accordingly.

\begin{verbatim}
\markup { \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\hspace*{1in} \hspace*{1in} \\
\end{verbatim}
\hspace \hspace amount (number)

Create an invisible object taking up horizontal space amount.
\markupspec
\hspace #2 \hspace #8

one two three
\justify-field \text{symbol} \quad \text{(symbol)}

Justify the data which has been assigned to \text{symbol}.

\header{
  \text{title} = "My title"
  \text{myText} = "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{
  \text{bookTitleMarkup} = \markup{
    \text{column} {
      \text{fill-line} { \fromproperty \text{#'header:title} }
      \text{null}
      \text{justify-field} { \fromproperty \text{#'header:myText} }
    } 
  }
}

\markup{
  \null
}

\text{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-line \text{args} \quad \text{(markup list)}

Put \text{markups} in a horizontal line of width \text{line-width}. The markups are spread to fill the entire line and separated by equal space. If there are no arguments, return an empty stencil.

\markup{
  \text{ justify-line }
  \text{Space between neighboring words is constant}
}

Space between neighboring words is constant

Used properties:
- line-width (#f)
- word-space (0.6)
- text-direction (1)
\justify args (markup list)

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 {
   \justfy {
      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.
\[\text{\texttt{\left-align \arg}} \text{ (markup)}\]
\[
\text{\texttt{\{ \texttt{\left-align \arg} <text> \} \}}
\]
\[
\text{one} \quad \text{two} \quad \text{three}
\]

`\left-column args` (markup list)
Put `args` in a left-aligned column.
\[\text{\texttt{\left-column \args}} \text{ (markup list)}\]
\[
\text{\texttt{\{ \texttt{\left-column \args} <text> \} \}}
\]
\[
\text{one} \quad \text{two} \quad \text{three}
\]

Used properties:
- baseline-skip

`\line args` (markup list)
Put `args` in a horizontal line. The property `word-space` determines the space between markups in `args`.
\[\text{\texttt{\line \args}} \text{ (markup list)}\]
\[
\text{\texttt{\{ \texttt{\line \args} <text> \} \}}
\]
\[
\text{one} \quad \text{two} \quad \text{three}
\]

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 {
  \markup {
    one
    \lower #3
    two
    three
  }
  one three
two
}

\overlay args (markup list)
Takes a list of markups combining them.
\markup {
  \markup {
    \fontsize #5
    \override #'(thickness . 2)
    \overlay {
      \draw-line #'(0 . 4)
      \arrow-head #Y #DOWN ##f
      \translate #'(0 . 4)\arrow-head #Y #UP ##f
    }
  }
}

\pad-around amount (number) arg (markup)
Add padding amount all around arg.
\markup {
  \markup {
    \box {
      \hspace #2
      \pad-around #0.5 {
        padded
      }
    }
  }
}

\pad-markup amount (number) arg (markup)
Add space around a markup object. Identical to pad-around.
\markup {
  \box {
    \hspace #2
  }
}
\pad-to-box \textit{x-ext} (pair of numbers) \textit{y-ext} (pair of numbers) \textit{arg} (markup)

Make \textit{arg} take at least \textit{x-ext}, \textit{y-ext} space.

\raise \textit{amount} (number) \textit{arg} (markup)

Raise \textit{arg} by the distance \textit{amount}. A negative \textit{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 \( \frac{9}{7} + \)

\right-align \ arg (markup)
Align \ arg \ on its right edge.
\markup {
  \markup {
    \column {
      one
      \right-align
      two
      three
    }
  }
  one
two
three

\right-column \ args (markup list)
Put \ args \ in a right-aligned column.
\markup {
  \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
\translate offset (pair of numbers) arg (markup)
Translate arg relative to its surroundings. offset is a pair of numbers representing the displacement in the X and Y axis.
\markup {
  * \translate #'(2 . 3)
  \line { translated two spaces right, three up }
}

translating 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
  }
}

translate-scaled

Used properties:
- font-size (0)

\vcenter arg (markup)
Align arg to its Y center.
\markup {
  one
  \vcenter
  two
  three
}

one two three

\vspace amount (number)
Create an invisible object taking up vertical space of amount multiplied by 3.
\markup {

Anhang A: Notationsübersicht

\center-column 
one \vspace #2
two \vspace #5
three 
}
}

one

two

three

\wordwrap-field symbol (symbol) 
Wordwrap the data which has been assigned to symbol.
\header 
\title = "My title"
\myText = "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:myText
}

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\ \texttt{args} (markup list)

Simple wordwrap. Use \texttt{\override \#'(line-width . X)} to set the line width, where \textit{X} is the number of staff spaces.

\begin{verbatim}
\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.
}
}
\end{verbatim}

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:

- \texttt{text-direction (1)}
- \texttt{word-space}
- \texttt{line-width (#f)}
- \texttt{baseline-skip}

\wordwrap-string\ \texttt{arg} (string)

Wordwrap a string. Paragraphs may be separated with double newlines.

\begin{verbatim}
\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"
}
\end{verbatim}
Lorem ipsum dolor sit amet, consectetur adipiscing 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

A.10.3 Graphic
\arrow-head axis (integer) dir (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
}
\circle arg (markup)

Draw a circle around \textit{arg}. Use \texttt{thickness}, \texttt{circle-padding} and \texttt{font-size} properties to determine line thickness and padding around the markup.

\begin{verbatim}
\markup {\circle {Hi}}
\end{verbatim}

Used properties:
- \texttt{circle-padding} (0.2)
- \texttt{font-size} (0)
- \texttt{thickness} (1)

\draw-circle radius (number) thickness (number) filled (boolean)

A circle of radius \textit{radius} and thickness \textit{thickness}, optionally filled.

\begin{verbatim}
\markup {\draw-circle #2 #0.5 ##f \hspace #2 \draw-circle #2 #0 ##t}
\end{verbatim}

\draw-dashed-line dest (pair of numbers)

A dashed line.

If \texttt{full-length} is set to \texttt{#t} (default) the dashed-line extends to the whole length given by \textit{dest}, without white space at beginning or end. \texttt{off} will then be altered to fit. To insist on the given (or default) values of on, off use \texttt{\override \#'(full-length . \#f)} Manual settings for on, off and phase are possible.

\begin{verbatim}
\markup {\draw-dashed-line #'(5.1 . 2.3) \override #'(on . 0.3) \override #'(off . 0.5) \draw-dashed-line #'(5.1 . 2.3)}
\end{verbatim}

Used properties:
- \texttt{full-length} (#t)
- \texttt{phase} (0)
- \texttt{off} (1)
• on (1)
• thickness (1)
\draw-dotted-line dest (pair of numbers)
A dotted line.
The dotted-line always extends to the whole length given by dest, without white space at beginning or end. Manual settings for off are possible to get larger or smaller space between the dots. The given (or default) value of off will be altered to fit the line-length.
\markup {
\draw-dotted-line #'(5.1 . 2.3)
\override #'(thickness . 2)
\override #'(off . 0.2)
\draw-dotted-line #'(5.1 . 2.3)
}

Used properties:
• phase (0)
• off (1)
• thickness (1)
\draw-hline
Draws a line across a page, where the property span-factor controls what fraction of the page is taken up.
\markup {
\column {
\draw-hline
\override #'(span-factor . 1/3)
\draw-hline
}
}

Used properties:
• span-factor (1)
• line-width
• draw-line-markup
\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)

\draw-squiggle-line sq-length (number) dest (pair of numbers) eq-end? (boolean)

A squiggled line.

If eq-end? is set to #t, it is ensured the squiggled line ends with a bow in the same direction as the starting one. sq-length is the length of the first bow. dest is the endpoint of the squiggled line. To match dest the squiggled line is scaled accordingly. Its appearance may be customized by overrides for thickness, angularity, height and orientation.

\markup {
\column {
\draw-squiggle-line #0.5 #'(6 . 0) ##t
\override #'(orientation . -1)
\draw-squiggle-line #0.5 #'(6 . 0) ##t
\draw-squiggle-line #0.5 #'(6 . 0) ##f
\override #'(height . 1)
\draw-squiggle-line #0.5 #'(6 . 0) ##t
\override #'(thickness . 5)
\draw-squiggle-line #0.5 #'(6 . 0) ##t
\override #'(angularity . 2)
\draw-squiggle-line #0.5 #'(6 . 0) ##t
}
\ellipse {
    Hi
}
}

Hi

Used properties:

- orientation (1)
- height (0.5)
- angularity (0)
- thickness (0.5)

\ellipse arg (markup)

Draw an ellipse around arg. Use thickness, x-padding, y-padding and font-size properties to determine line thickness and padding around the markup.

\markup {
    \ellipse {
        Hi
    }
}

Hi

Used properties:

- y-padding (0.2)
- x-padding (0.2)
\begin{itemize}
\item font-size (0)
\item thickness (1)
\end{itemize}

**\texttt{\textbackslash epsfile axis (number) size (number) file-name (string)}**

Inline an EPS image. The image is scaled along axis to size.

\begin{verbatim}
\markup {
\general-align #Y #DOWN {
  \epsfile #X #20 #'"context-example.eps"
  \epsfile #Y #20 #'"context-example.eps"
}
}
\end{verbatim}

**\texttt{\textbackslash 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,

\begin{verbatim}
\markup {
  \filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0
}
\end{verbatim}

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).

\begin{verbatim}
\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
}
\end{verbatim}

**\texttt{\textbackslash hbracket arg (markup)}**

Draw horizontal brackets around arg.

\begin{verbatim}
\markup {
  \hbracket {
    \line {
      one two three
    }
  }
}
\end{verbatim}

\begin{verbatim}
one two three
\end{verbatim}
\oval arg (markup)

Draw an oval around arg. Use \texttt{thickness}, \texttt{x-padding}, \texttt{y-padding} and \texttt{font-size} properties to determine line thickness and padding around the markup.

\begin{verbatim}
\markup {
  \oval {
    Hi
  }
}
\end{verbatim}

Hi

Used properties:

- \texttt{y-padding} (0.75)
- \texttt{x-padding} (0.75)
- \texttt{font-size} (0)
- \texttt{thickness} (1)

\parenthesize arg (markup)

Draw parentheses around arg. This is useful for parenthesizing a column containing several lines of text.

\begin{verbatim}
\markup {
  \line {
    \parenthesize {
      \column {
        foo
        bar
      }
    }
    \override #'(angularity . 2) {
      \parenthesize {
        \column {
          bah
          baz
        }
      }
    }
  }
}
\end{verbatim}

\begin{tabular}{|c|c|}
\hline
foo & bah \\
\hline
bar & baz \\
\hline
\end{tabular}

Used properties:

- \texttt{width} (0.25)
- \texttt{line-thickness} (0.1)
- \texttt{thickness} (1)
- \texttt{size} (1)
- \texttt{padding}
- \texttt{angularity} (0)
\path thickness (number) commands (list)

Draws a path with line thickness according to the directions given in commands. commands is a list of lists where the car of each sublist is a drawing command and the cdr comprises the associated arguments for each command.

There are seven commands available to use in the list commands: moveto, rmoveto, lineto, rlineto, curveto, rcurveto, and closepath. Note that the commands that begin with r are the relative variants of the other three commands.

The commands moveto, rmoveto, lineto, and rlineto take 2 arguments; they are the X and Y coordinates for the destination point.

The commands curveto and rcurveto create cubic Bézier curves, and take 6 arguments; the first two are the X and Y coordinates for the first control point, the second two are the X and Y coordinates for the second control point, and the last two are the X and Y coordinates for the destination point.

The closepath command takes zero arguments and closes the current subpath in the active path.

Note that a sequence of commands must begin with a moveto or rmoveto to work with the SVG output.

Line-cap styles and line-join styles may be customized by overriding the line-cap-style and line-join-style properties, respectively. Available line-cap styles are 'butt, 'round, and 'square. Available line-join styles are 'miter, 'round, and 'bevel.

The property filled specifies whether or not the path is filled with color.

samplePath =
#'((moveto 0 0)
  (lineto -1 1)
  (lineto 1 1)
  (lineto 1 -1)
  (curveto -5 -5 -5 5 -1 0)
  (closepath))

\markup {
  \path #0.25 #samplePath

  \override #'(line-join-style . miter) \path #0.25 #samplePath

  \override #'(filled . #t) \path #0.25 #samplePath
}

Used properties:
• filled (#f)
• line-join-style (round)
• line-cap-style (round)

\postscript str (string)
This inserts str directly into the output as a PostScript command string.

ringsps = #"0.15 setlinewidth
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).

\scale factor-pair (pair of numbers) arg (markup)
Scale arg. factor-pair is a pair of numbers representing the scaling-factor in the X and Y axes. Negative values may be used to produce mirror images.

\markup {\scale #'(2 . 1) stretched}\scale #'(1 . -1)
\triangle \text{ filled (boolean)}
A triangle, either filled or empty.

\begin{verbatim}
\markup {
  \triangle \text{##t}
  \hspace #2
  \triangle \text{##f}
}
\end{verbatim}

\begin{itemize}
  \item baseline-skip (2)
  \item font-size (0)
  \item thickness (0.1)
\end{itemize}

\with-url \text{url (string) arg (markup)}
Add a link to URL \text{url} around \text{arg}. This only works in the PDF backend.

\begin{verbatim}
\markup {
  \with-url #'"http://lilypond.org/" {
    LilyPond ... \text{italic} {
      music notation for everyone
    }
  }
}
\end{verbatim}

\text{LilyPond ... music notation for everyone}

\subsection{A.10.4 Music}

\compound-meter \text{time-sig (number or pair)}
Draw a numeric time signature.

\begin{verbatim}
\markup {
  \column {
    \line { \text{Single number: \compound-meter #3 } }
    \line { Conventional: \compound-meter #'(4 . 4)
              or \compound-meter #'(4 4) }
    \line { Compound: \compound-meter #'(2 3 8) }
    \line { Single-number compound: \compound-meter #'((2) (3)) }
    \line { Complex compound: \compound-meter #'((2 3 8) (3 4)) }
  }
}
\end{verbatim}
Anhang A: Notationsübersicht

Single number: 3
Conventional: \( \frac{3}{4} \) or \( \frac{3}{4} \)
Compound: \( \frac{2 \cdot 3}{8} \)
Single-number compound: \( 2 + 3 \)
Complex compound: \( 2 + \frac{3}{8} + \frac{3}{4} \)

\( \texttt{\textbackslash customTabClef \textbackslash num-strings (integer) \textbackslash staff-space (number)} \)
Draw a tab clef sans-serif style.

\( \texttt{\textbackslash doubleflat} \)
Draw a double flat symbol.

\[ \texttt{\markup { \textbackslash doubleflat }} \]
\( \texttt{\textbackslash doubleflat} \)

\( \texttt{\textbackslash doublesharp} \)
Draw a double sharp symbol.

\[ \texttt{\markup { \textbackslash doublesharp }} \]
\( \texttt{\textbackslash doublesharp} \)

\( \texttt{\textbackslash fermata} \)
Create a fermata glyph. When \textit{direction} is \texttt{DOWN}, use an inverted glyph. Note that within music, one would usually use the \texttt{\textbackslash fermata} articulation instead of a markup.

\[ \texttt{\markup { \textbackslash fermata \textbackslash override \#(direction . ,DOWN) \textbackslash fermata }} \]
\( \texttt{\textbackslash fermata \textbackslash override \#(direction . ,DOWN) \textbackslash fermata} \)

\( \texttt{\textbackslash flat} \)
Draw a flat symbol.

\[ \texttt{\markup { \textbackslash flat }} \]
\( \texttt{\flat} \)

\( \texttt{\textbackslash musicglyph \textbackslash glyph-name (string)} \)
\textit{glyph-name} is converted to a musical symbol; for example, \texttt{\musicglyph \#"accidentals.natural"} selects the natural sign from the music font. See
Abschnitt “The Emmentaler font” in Notationsreferenz for a complete listing of the possible glyphs.

\markup {
\musicglyph #"f"
\musicglyph #"rests.2"
\musicglyph #"clefs.G_change"
}

\natural

Draw a natural symbol.

\markup {
\natural
}

\note-by-number log (number) dot-count (number) dir (number)

Construct a note symbol, with stem and flag. By using fractional values for dir, longer or shorter stems can be obtained. Supports all note-head-styles. Ancient note-head-styles will get mensural-style-flags. flag-style may be overridden independently. Supported flag-styles are default, old-straight-flag, modern-straight-flag, flat-flag, mensural and neomensural. The latter two flag-styles will both result in mensural-flags. Both are supplied for convenience.

\markup {
\note-by-number #3 #0 #DOWN
\hspace #2
\note-by-number #1 #2 #0.8
}

\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
\rest-by-number log (number) dot-count (number)
A rest or multi-measure-rest symbol.
\markup {
    \rest-by-number #3 #2
    \hspace #2
    \rest-by-number #0 #1
    \hspace #2
    \override #'(multi-measure-rest . #t)
    \rest-by-number #0 #0
}

\rest duration (string)
This produces a rest, with the duration for the rest type and augmentation dots.
"breve", "longa" and "maxima" are valid input-strings.
Printing MultiMeasureRests could be enabled with \override #'(multi-measure-rest . #t) If MultiMeasureRests are taken, the MultiMeasureRestNumber is printed above. This is enabled for all styles using default-glyphs. Could be disabled with \override #'(multi-measure-rest-number . #f)
\markup {
    \rest #'"4.."
    \hspace #2
    \rest #'"breve"
    \hspace #2
    \override #'(multi-measure-rest . #t)
    { \rest #'"7"
    \hspace #2
    \override #'(multi-measure-rest-number . #f)
    \rest #'"7"
    }
}

Used properties:
• style ("")
• flag-style ("")
• font-size (0)
• multi-measure-rest-number (#t)

• multi-measure-rest (#f)

• style (')())

\score score (score)

Inline an image of music. The reference point (usually the middle staff line) of the lowest staff in the top system is placed on the baseline.

\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
  }
}
Allegro

Used properties:
- baseline-skip

\semiflat
Draw a semiflat symbol.
\markup { \semiflat }
\f

\semisharp
Draw a semisharp symbol.
\markup { \semisharp }
\f

\sesquiflat
Draw a 3/2 flat symbol.
\markup { \sesquiflat }
\f

\sesquisharp
Draw a 3/2 sharp symbol.
\markup { \sesquisharp }
\f

\sharp
Draw a sharp symbol.
\markup { \sharp }
\f

\tied-lyric str (string)
Like simple-markup, but use tie characters for \("\) tilde symbols.
\markup \column {
Siam navi all’onde algenti Lasciate in abbandono
Impetuosi venti I nostri affetti sono
Ogni diletto e scoglio Tutta la vita e un mar.

Used properties:
• word-space

A.10.5 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 (relative to normal line thickness). Default: 0.5.
  • 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 indications 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 \markup { marking-list (pair) }

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.

- (capo fret-number)
  Place a capo indicator (a large solid bar) across the entire fretboard at fret location fret-number. Also, set fret fret-number to be the lowest fret on the fret diagram.

  (place-fret string-number fret-number [finger-value] [color-modifier] [color] \[parenthesized \[default-paren-color\]])

  Place a fret playing indication on string string-number at fret fret-number with an optional fingering label finger-value, an optional color modifier color-modifier, an optional color color, an optional parenthesis \[parenthesized\] and an optional paranthesis color \[default-paren-color\]. By default, the fret playing indicator is a solid dot. This can be globally changed by setting the value of
the variable dot-color or for a single dot by setting the value of color. The dot can be parenthesized by adding 'parenthesized. By default the color for the parenthesis is taken from the dot. Adding 'default-paren-color will take the parenthesis-color from the global dot-color, as a fall-back black will be used. Setting color-modifier to inverted inverts the dot color for a specific fingering. The values for string-number, fret-number, and the optional finger should be entered first in that order. The order of the other optional arguments does not matter. 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 (1.2)

\woodwind-diagram instrument (symbol) user-draw-commands (list)

Make a woodwind-instrument diagram. For example, say

\markup \woodwind-diagram
   #'oboe #'((lh . (d ees)) (cc . (five3qT1q)) (rh . (gis)))

for an oboe with the left-hand d key, left-hand ees key, and right-hand gis key
depressed while the five-hole of the central column effectuates a trill between 1/4
and 3/4 closed.

The following instruments are supported:
- piccolo
- flute
- oboe
- clarinet
- bass-clarinet
- saxophone
- bassoon
- contrabassoon

To see all of the callable keys for a given instrument, include the function
(print-keys 'instrument) in your .ly file, where instrument is the instrument
whose keys you want to print.

Certain keys allow for special configurations. The entire gamut of configurations
possible is as follows:
- 1q (1/4 covered)
- 1h (1/2 covered)
- 3q (3/4 covered)
- R (ring depressed)
- F (fully covered; the default if no state put)

Additionally, these configurations can be used in trills. So, for example, three3qTR
effectuates a trill between 3/4 full and ring depressed on the three hole. As another
example, threeRT effectuates a trill between R and open, whereas threeTR effectuates
a trill between open and shut. To see all of the possibilities for all of the keys
of a given instrument, invoke (print-keys-verbose 'instrument).

Lastly, substituting an empty list for the pressed-key alist will result in a diagram
with all of the keys drawn but none filled, for example:

\markup \woodwind-diagram #'oboe #'()

Used properties:
• graphical (#t)
• thickness (0.1)
• size (1)

A.10.6 Accordion Registers

\discant name (string)

\discant name generates a discant accordion register symbol.
To make it available,

```scheme
#(use-modules (scm accreg))
```
is required near the top of your input file.

The register names in the default \texttt{discant} register set have modeled after numeric Swiss notation like depicted in \url{http://de.wikipedia.org/wiki/Register_%28Akkordeon%29}, omitting the slashes and dropping leading zeros.

The string \texttt{name} is basically a three-digit number with the lowest digit specifying the number of 16' reeds, the tens the number of 8' reeds, and the hundreds specifying the number of 4' reeds. Without modification, the specified number of reeds in 8' is centered in the symbol. Newer instruments may have registrations where 8' can be used either within or without a tone chamber, \textquote{cassotto}. Notationally, the central dot then indicates use of cassotto. One can suffix the tens' digits \textquote{1} and \textquote{2} with \textquote{+} or \textquote{-} to indicate clustering the dots at the right or left respectively rather than centered.

Some examples are

\begin{verbatim}
\discant "1"  \discant "1+0"
\discant "120"  \discant "131"
\end{verbatim}

Used properties:
\begin{itemize}
\item \texttt{font-size} (0)
\end{itemize}

\texttt{freeBass name} (string)
\texttt{freeBass name} generates a free bass/converter accordion register symbol for the usual two-reed layout.

To make it available,

```scheme
#(use-modules (scm accreg))
```
is required near the top of your input file.

Available registrations are

\begin{verbatim}
\freeBass "1"  \freeBass "11"
\freeBass "10"
\end{verbatim}

Used properties:
\begin{itemize}
\item \texttt{font-size} (0)
\end{itemize}

\texttt{stdBass name} (string)
\texttt{stdBass name} generates a standard bass accordion register symbol.

To make it available,

```scheme
#(use-modules (scm accreg))
```
is required near the top of your input file.
The default bass register definitions have been modeled after the article http://www.accordions.com/index/art/stradella.shtml originally appearing in Accord Magazine.

The underlying register model is

![Diagram of bass register model]

This kind of overlapping arrangement is common for Italian instruments though the exact location of the octave breaks differ.

When not composing for a particular target instrument, using the five reed definitions makes more sense than using a four reed layout: in that manner, the 'Master' register is unambiguous. This is rather the rule in literature bothering about bass registrations at all.

Available registrations are

\begin{align*}
\texttt{\stdBass "Soprano"} & \quad \texttt{\stdBass "Soft Bass"} \\
\texttt{\stdBass "Alto"} & \quad \texttt{\stdBass "Soft Tenor"} \\
\texttt{\stdBass "Tenor"} & \quad \texttt{\stdBass "Bass/Alto"} \\
\texttt{\stdBass "Master"} & \quad \texttt{\stdBass "Soft Bass"}\end{align*}

Used properties:
- \texttt{font-size (0)}

\texttt{\stdBassIV name (string)}

\texttt{\stdBassIV name} generates a standard bass accordion register symbol.

To make it available,

\begin{verbatim}
#(use-modules (scm accreg))
\end{verbatim}

is required near the top of your input file.

The main use is for four-reed standard bass instruments with reedbank layout.
Notable instruments are Morino models with MIII (the others are five-reed instead) and the Atlantic IV. Most of those models have three register switches. Some newer Morinos with MIII might have five or even seven.

The prevalent three-register layout uses the middle three switches ‘Tenor’, ‘Master’, ‘Soft Bass’. Note that the sound is quite darker than the same registrations of ‘c’/-based instruments.

Available registrations are

\stdBassIV "Soprano" \stdBassIV "Soft Bass"
\stdBassIV "Alto" \stdBassIV "Bass/Alto"
\stdBassIV "Tenor" \stdBassIV "Soft Bass/Alto"
\stdBassIV "Master" \stdBassIV "Soft Tenor"

Used properties:
- font-size (0)

\stdBassV name (string)
\stdBassV name generates a standard bass accordion register symbol.

To make it available,

#(use-modules (scm accreg))

is required near the top of your input file.

The main use is for five-reed standard bass instruments with reedbank layout

This tends to be the bass layout for Hohner’s Morino series without convertor or MIII manual.
With the exception of the rather new 7-register layout, the highest two chord reeds are usually sounded together. The Older instruments offer 5 or 3 bass registers. The Tango VM offers an additional ‘Solo Bass’ setting that mutes the chord reeds. The symbol on the register buttons of the Tango VM would actually match the physical five-octave layout reflected here, but it is not used in literature.

Composers should likely prefer the five-reed versions of these symbols. The mismatch of a four-reed instrument with five-reed symbols is easier to resolve for the player than the other way round.

Available registrations are

\begin{verbatim}
\stdBassVI "Bass/Alto" \stdBassVI "Soft Bass"
\stdBassVI "Soft Bass/Alto" \stdBassVI "Soft Tenor"
\stdBassVI "Alto" \stdBassVI "Soprano"
\stdBassVI "Tenor" \stdBassVI "Sopranos"
\stdBassVI "Master" \stdBassVI "Solo Bass"
\end{verbatim}

Used properties:
• \texttt{font-size (0)}

\stdBassVI \texttt{name} (string)
\stdBassVI \texttt{name} generates a standard bass accordion register symbol for six reed basses.

To make it available,

\begin{verbatim}
#(use-modules (scm accreg))
\end{verbatim}

is required near the top of your input file.

This is primarily the register layout for the Hohner „Gola“ model. The layout is

\begin{verbatim}
\end{verbatim}

The registers are effectively quite similar to that of \stdBass. An additional bass reed at alto pitch is omitted for esthetical reasons from the ‘Master’ setting, so the
symbols are almost the same except for the ‘Alto/Soprano’ register with bass notes at Alto pitch and chords at Soprano pitch.

Available registrations are

\stdBassVI "Soprano" \stdBassVI "Alto/Soprano"
\stdBassVI "Alto" \stdBassVI "Bass/Alto"
\stdBassVI "Soft Tenor" \stdBassVI "Soft Bass"
\stdBassVI "Master"

Used properties:
• font-size (0)

A.10.7 Other
\auto-footnote mkup (markup) note (markup)
Have footnote note act as an annotation to the markup mkup.
\markup {
    \auto-footnote a b
    \override #'(padding . 0.2)
    \auto-footnote c d
}
\auto-footnote a c

The footnote will be annotated automatically.

Used properties:
• padding (0.0)
• raise (0.5)

\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
}
\backslashed-digit 5 7

Used properties:
• thickness (1.6)
• font-size (0)

\char \texttt{num} \texttt{(integer)}
Produce a single character. Characters encoded in hexadecimal format require the
prefix \#x.

\markup \{ \char \#65 \char \#x00a9 \}

A ©

\eyeglasses
Prints out eyeglasses, indicating strongly to look at the conductor.

\markup { \eyeglasses }

\first-visible \texttt{args} \texttt{(markup list)}
Use the first markup in \texttt{args} that yields a non-empty stencil and ignore the rest.

\markup \{ \first-visible { \fromproperty #'header:composer \italic Unknown } \}

\footnote \texttt{mkup} \texttt{(markup)} \texttt{note} \texttt{(markup)}
Have footnote \texttt{note} act as an annotation to the markup \texttt{mkup}.

\markup \{ \auto-footnote a b \override #'(padding . 0.2) \auto-footnote c d \}

a c

The footnote will not be annotated automatically.

\fraction \texttt{arg1} \texttt{(markup)} \texttt{arg2} \texttt{(markup)}
Make a fraction of two markups.

\markup \{ \pi \approx \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

\left-brace size (number)
A feta brace in point size size.
\markup {
  \left-brace #35
  \hspace #2
  \left-brace #45
}

\lookup glyph-name (string)
Look up a glyph by name.
\markup {
  \override #'(font-encoding . fetaBraces) {
    \lookup #"brace200"
    \hspace #2
    \rotate #180
    \lookup #"brace180"
  }
}


\markalphabet \texttt{num} (integer)
Make a markup letter for \texttt{num}. The letters start with A to Z and continue with double letters.
\begin{verbatim}
\markup {
  \markalphabet #8
  \hspace #2
  \markalphabet #26
}
\end{verbatim}

I AA

\markletter \texttt{num} (integer)
Make a markup letter for \texttt{num}. The letters start with A to Z (skipping letter I), and continue with double letters.
\begin{verbatim}
\markup {
  \markletter #8
  \hspace #2
  \markletter #26
}
\end{verbatim}

J AB

\null
An empty markup with extents of a single point.
\begin{verbatim}
\markup {
  \null
}
\end{verbatim}

\on-the-fly \texttt{procedure} (procedure) \texttt{arg} (markup)
Apply the \texttt{procedure} markup command to \texttt{arg}. \texttt{procedure} takes the same arguments as \texttt{interpret-markup} and returns a stencil.

\override \texttt{new-prop} (pair) \texttt{arg} (markup)
Add the argument \texttt{new-prop} to the property list. Properties may be any property supported by Abschnitt “font-interface” in Referenz der Interna, Abschnitt “text-interface” in Referenz der Interna and Abschnitt “instrument-specific-markup-interface” in Referenz der Interna.
\begin{verbatim}
\markup {
  \line {
    \column {
      default
      baseline-skip
    }
    \hspace #2
    \override #'(baseline-skip . 4) {
      \column {
        increased
        baseline-skip
      }
    }
  }
}\end{verbatim}
\page-link \textit{page-number} (number) \textit{arg} (markup)
Add a link to the page \textit{page-number} around \textit{arg}. This only works in the PDF backend.
\begin{verbatim}
\markup {
  \page-link #2 { \textit{This links to page 2...} }
}
\end{verbatim}

This links to page 2...

\page-ref \textit{label} (symbol) \textit{gauge} (markup) \textit{default} (markup)
Reference to a page number. \textit{label} is the label set on the referenced page (using the \label command), \textit{gauge} a markup used to estimate the maximum width of the page number, and \textit{default} the value to display when \textit{label} is not found.
(If the current book or bookpart is set to use roman numerals for page numbers, the reference will be formatted accordingly – in which case the \textit{gauge}’s width may require additional tweaking.)

\pattern \textit{count} (integer) \textit{axis} (integer) \textit{space} (number) \textit{pattern} (markup)
Prints \textit{count} times a \textit{pattern} markup. Patterns are spaced apart by \textit{space}. Patterns are distributed on \textit{axis}.
\begin{verbatim}
\markup \column {
  "Horizontally repeated :
  \pattern #7 #X #2 \flat
  \null
  "Vertically repeated :
  \pattern #3 #Y #0.5 \flat
}
\end{verbatim}

Horizontally repeated:
\begin{verbatim}
\textbf{b} \textbf{b} \textbf{b} \textbf{b} \textbf{b} \textbf{b}
\end{verbatim}

Vertically repeated:
\begin{verbatim}
\textbf{b}
\textbf{b}
\end{verbatim}

\property-recursive \textit{symbol} (symbol)
Print out a warning when a header field markup contains some recursive markup definition.

\right-brace \textit{size} (number)
A fetta brace in point size \textit{size}, rotated 180 degrees.
\begin{verbatim}
\markup {
  \right-brace #45
  \hspace #2
}\end{verbatim}
\right-brace #35
}

\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 (stencil)
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 \arg transparent.
\markup {
\transparent {
invisible text
}
}

\verbatim-file \name (string)
Read the contents of file \name, and include it verbatim.
\markup {
\verbatim-file #"simple.ly"
}
%% A simple piece in LilyPond, a scale.
\relative { 
c' d e f g a b c 
} 

%% Optional helper for automatic updating by convert-ly.
%% May be omitted.
\version "2.19.21"

\whiteout arg (markup)
Provide a white background for arg. The shape of the white background is determined by style. The default is box which produces a rectangle. rounded-box produces a rounded rectangle. outline approximates the outline of the markup.

\markup {
\combine \filled-box #'(-1 . 15) #'(-3 . 4) #1
\override #'(thickness . 1.5)
\whiteout whiteout-box
}
\markup {
\combine \filled-box #'(-1 . 24) #'(-3 . 4) #1
\override #'(style . rounded-box)
\override #'(thickness . 3)
\whiteout whiteout-rounded-box
}
\markup {
\combine \filled-box #'(-1 . 18) #'(-3 . 4) #1
\override #'(style . outline)
\override #'(thickness . 3)
\whiteout whiteout-outline
}

\whiteout-box
\whiteout-rounded-box
\whiteout-outline

Used properties:
• thickness ('())
• style (box)

\with-color color (color) arg (markup)
Draw arg in color specified by color.

\markup {
\with-color #red
red
A.11 Textbeschriftungslistenbefehle

Folgende Befehle können mit dem Befehl \markuplist zusammen benutzt werden:

\column-lines args (markup list)
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 (markup list)
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

\map-markup-commands compose (procedure) args (markup list)
This applies the function compose to every markup in args (including elements of markup list command calls) in order to produce a new markup list. Since the return value from a markup list command call is not a markup list but rather a list of
stencils, this requires passing those stencils off as the results of individual markup calls. That way, the results should work out as long as no markups rely on side effects.

\override-lines new-prop (pair) args (markup list)
Like \override, for markup lists.

\score-lines score (score)
This is the same as the \score markup but delivers its systems as a list of lines. Its score argument is entered in braces like it would be for \score.

\table column-align (number list) lst (markup list)
Returns a table.

\markuplist {
  \override #'(padding . 2)
  \table
  #'(0 1 0 -1)
  {
    \underline { center-aligned right-aligned center-aligned left-aligned }
    one \number 1 thousandth \number 0.001
    eleven \number 11 hundredth \number 0.01
    twenty \number 20 tenth \number 0.1
    thousand \number 1000 one \number 1.0
  }
}

center-aligned  right-aligned  center-aligned  left-aligned

one  \underline { \number 1 thousandth \number 0.001 }

eleven  \underline { \number 11 hundredth \number 0.01 }

twenty  \underline { \number 20 tenth \number 0.1 }

thousand  \underline { \number 1000 one \number 1.0 }

Used properties:
• baseline-skip
• padding (0)

\table-of-contents
Used properties:
• baseline-skip
\wordwrap-internal justify (boolean) args (markup list)
Internal markup list command used to define \justify and \wordwrap.
Used properties:
- text-direction (1)
- word-space
- line-width (#f)

\wordwrap-lines args (markup list)
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

A.12 Liste der Sonderzeichen
Folgende Sonderezeichen-Bezeichnungen können benutzt werden, zu mehr Details siehe [ASCII-Aliase], Seite 484.

die HTML-Syntax wird benutzt und die meisten der Bezeichnungen sind die gleichen wie für HTML. Der Rest ist durch \LaTeX inspiriert.


&hellip; · &ndash; · &mdash; · &iexcl; ·
&iquest; · &sol; &solidus; · &flq; · &frq; ·
&flqq; · &frqq; · &glq; · &grq; ·
&glqq; · &grqq; · &elq; · &erq; ·
&elqq; · &erqq; · &ensp; · &emsp; ·
&thinsp; · &nbsp; · &nnb; · &zwj; ·
&zwj; · &middot; · &bull; · &copyright; ·
A.13 Liste der Artikulationszeichen

Die Skripte unten sind in der Feta-Glyphe definiert und können an Noten angehängt werden (etwa ‘c\accent’).

Artikulationsskripte

\begin{center}
\begin{tabular}{c c c c}
\texttt{accent} & \texttt{espressivo} & \texttt{marcato} & \texttt{portato} \\
\texttt{staccatissimo} & \texttt{staccato} & \texttt{tenuto} \\
\end{tabular}
\end{center}

Ornamentale Skripte

\begin{center}
\begin{tabular}{c c c c c}
\texttt{prall} & \texttt{mordent} & \texttt{prallmordent} & \texttt{turn} & \texttt{upprall} \\
\texttt{downprall} & \texttt{upmordent} & \texttt{downmordent} & \texttt{lineprall} \\
\texttt{prallprall} & \texttt{pralldown} & \texttt{prallup} & \texttt{reverseturn} & \texttt{trill} \\
\end{tabular}
\end{center}

Fermatenskripte

\begin{center}
\begin{tabular}{c c c c c}
\texttt{shortfermata} & \texttt{fermata} & \texttt{longfermata} & \texttt{verylongfermata} \\
\end{tabular}
\end{center}

Instrumentenspezifische Skripte

\begin{center}
\begin{tabular}{c c c c c}
\texttt{upbow} & \texttt{downbow} & \texttt{flageolet} & \texttt{snappizzicato} \\
\texttt{open} & \texttt{halfopen} & \texttt{halfopenvertical} & \texttt{stopped} \\
\texttt{lheel} & \texttt{rheel} & \texttt{ltoe} & \texttt{rtoe} \\
\end{tabular}
\end{center}
Wiederholungszeichen und Scripts

Ancient scripts

A.14 Schlagzeugnoten
<table>
<thead>
<tr>
<th>Notation</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>mutehiconga: cghm</td>
<td></td>
</tr>
<tr>
<td>openhiconga: cgho</td>
<td></td>
</tr>
<tr>
<td>openlocona: cglo</td>
<td></td>
</tr>
<tr>
<td>mutelocona: cglm</td>
<td></td>
</tr>
<tr>
<td>hicona: cgh</td>
<td></td>
</tr>
<tr>
<td>lococona: cgl</td>
<td></td>
</tr>
<tr>
<td>hitimbale: timh</td>
<td></td>
</tr>
<tr>
<td>hiagogo: agh</td>
<td></td>
</tr>
<tr>
<td>lotimbale: timl</td>
<td></td>
</tr>
<tr>
<td>loagogo: agl</td>
<td></td>
</tr>
<tr>
<td>hisidestick: ssh</td>
<td></td>
</tr>
<tr>
<td>losidestick: ssl</td>
<td></td>
</tr>
<tr>
<td>sidestick: ss</td>
<td></td>
</tr>
<tr>
<td>shortguiro: guis</td>
<td></td>
</tr>
<tr>
<td>guiro: gui</td>
<td></td>
</tr>
<tr>
<td>maracas: mar</td>
<td></td>
</tr>
<tr>
<td>longguiro: guil</td>
<td></td>
</tr>
<tr>
<td>cabasa: cab</td>
<td></td>
</tr>
<tr>
<td>shortwhistle: whs</td>
<td></td>
</tr>
<tr>
<td>longwhistle: whl</td>
<td></td>
</tr>
<tr>
<td>handclap: hc</td>
<td></td>
</tr>
<tr>
<td>vibraslap: vibs</td>
<td></td>
</tr>
<tr>
<td>tambourine: tamb</td>
<td></td>
</tr>
<tr>
<td>tamtam: tt</td>
<td></td>
</tr>
<tr>
<td>claves: cl</td>
<td></td>
</tr>
<tr>
<td>lowwoodblock: wbl</td>
<td></td>
</tr>
<tr>
<td>hiwoodblock: wbh</td>
<td></td>
</tr>
<tr>
<td>mutecuica: cuim</td>
<td></td>
</tr>
<tr>
<td>mutetriangle: trim</td>
<td></td>
</tr>
<tr>
<td>opentriangle: trio</td>
<td></td>
</tr>
<tr>
<td>opencuica: cuio</td>
<td></td>
</tr>
<tr>
<td>triangle: tri</td>
<td></td>
</tr>
<tr>
<td>oneup: ua</td>
<td></td>
</tr>
<tr>
<td>threeup: uc</td>
<td></td>
</tr>
<tr>
<td>fiveup: ue</td>
<td></td>
</tr>
<tr>
<td>twoup: ub</td>
<td></td>
</tr>
<tr>
<td>fourup: ud</td>
<td></td>
</tr>
<tr>
<td>onedown: da</td>
<td></td>
</tr>
<tr>
<td>threedown: dc</td>
<td></td>
</tr>
<tr>
<td>fivedown: de</td>
<td></td>
</tr>
<tr>
<td>twodown: db</td>
<td></td>
</tr>
<tr>
<td>fourdown: dd</td>
<td></td>
</tr>
</tbody>
</table>
A.15 Technisches Glossar

Ein Glossar der technischen Ausdrücke und Konzepte, die von LilyPond intern benutzt werden. Die Ausdrücke kommen in den Handbüchern, auf den Mailinglisten oder im Quellcode vor.

alist

Eine assoziative Liste oder **alist** in kurz ist ein Scheme-Paar, das einen Wert mit einem Schlüssel assoziiert: (Schlüssel . Wert). In der Datei `scm/lily.scm` beispielsweise assoziiert die alist „type-p-name-alist“ bestimmte Prädikate (etwa ly: music?) mit Bezeichnungen (wie „music“) sodass Fehler der Typüberprüfung über eine Konsolennachricht mitgeteilt werden können, die auch die Bezeichnung des erwarteten Typprädikats mitteilt.

callback

Ein **callback** ist eine Routine, Funktion oder Methode, deren Referenz in einem Aufruf als Argument an eine andere Routine weitergereicht wird, sodass die aufgerufene Routine ermöglicht wird, das Argument zu aktivieren. Die Technik ermöglicht es einer niedrigeren Ebene des Programmes, eine Funktion aufzurufen, die auf höherer Ebene definiert wurde. Callbacks werden sehr ausgiebig in LilyPond eingesetzt, um es Scheme-Code auf der Benutzerebene zu erlauben, wie viele Funktionen der niedrigeren Ebene ausgeführt werden sollen.

closure

In Scheme entsteht ein **closure** (Abschluss), wenn eine Funktion, normalerweise ein Lambda-Ausdruck, als Variable weitergegeben wird. Das closure enthält den Code der Funktion plus Verweise zu den lexikalischen Verknüpfungen der freien Variablen der Funktion (also die Variablen, die in Ausdrücken benutzt werden, aber außerhalb von ihnen definiert werden). Wenn diese Funktion später einem anderen Argument zugewiesen wird, werden die freien Variabel-Verknüpfungen, die in das closure eingeschlossen sind, benutzt um die Werte der freien Variablen, die in der Rechnung benutzt werden sollen, zu errechnen. Eine nützliche Eigenschaft von closures ist, dass man interne variable Werte zwischen den Aufrufen wiederverwerten kann, sodass ein Status erhalten bleiben kann.

Ein **simple closure** (einfacher Abschluss) ist ein closure, dessen Ausdruck keine freien Variablen und auch keine freien Variablel-Verknüpfungen hat.

Ein simple closure wird in LilyPond von einem smob dargestellt, der den Ausdruck und eine Methode, wie der Ausdruck auf eine Liste von Argumenten angewendet werden soll, enthält.

glyph

Ein **glyph** (Glyphe) ist eine bestimmte graphische Repräsentation eines typographischen Characters oder einer Kombination von zwei oder mehr Charakteren, die dann eine Ligatur bilden. Eine Gruppe an Glyphen des gleichen Stils bilden ein Font, und eine Gruppe an Fonts, die mehrere Stile darstellen, bilden eine Schriftfamilie (engl. typeface).

Siehe auch

Notationsreferenz: ⟨undefined⟩ [Fonts], Seite ⟨undefined⟩, Abschnitt 3.3.3 [Sonderzeichen], Seite 482.

grob

LilyPond-Objekte, die Elemente der Notation in der graphischen Ausgabe des Programmen darstellen, wie etwa Notenköpfe, Hälse, Bögen, Bindebögen, Fingersatz, Schlüssel usw., werden „Layout-Objekte“ genannt, auch oft als „GRaphische OBjekte“ bezeichnet, was dann zu grob abgekürzt wird.
Siehe auch


Referenz der Interna: Abschnitt “All layout objects” in Referenz der Interna.

immutable
Ein immutable (unberührbares) Objekt ist ein, dessen Status nach der Erstellung nicht mehr verändert werden kann, entgegen einem mutable Objekt, das nach der Erstellung noch verändert werden kann.

In LilyPond sind unberührbare oder geteilte Eigenschaften das Standardverhalten von Grobs. Sie werden zwischen vielen Objekten geteilt. Entgegen ihrer Bezeichnung können sie jedoch mit \override und \revert verändert werden.

Siehe auch
Notationsreferenz: [mutable], Seite 709.

interface
Aktionen und Eigenschaften, die eine Gruppe von Grobs gemeinsam haben, werden in ein Objekt gesammelt, das als grob-interface oder auch „Schnittstelle“ (engl. interface) bezeichnet wird.

Siehe auch

Notationsreferenz: Abschnitt 5.2.2 [Layout-Schnittstellen], Seite 569.

Referenz der Interna: Abschnitt “Graphical Object Interfaces” in Referenz der Interna.

lexer

mutable
Ein mutable (veränderbares) Objekt ist ein Objekt, dessen Status verändert werden kann, im Gegenteil zu einem immutable Objekt, dessen Status zur Entstehungszeit festgelegt ist.


Siehe auch
Notationsreferenz: [immutable], Seite 709.

output-def
Eine Instanz der Output-def-Klasse enthält die Methoden und Datenstruktur, die mit einem Ausgabebereich assoziiert wird. Instanzen werden für midi, layout und paper-Umgebungen erstellt.

**parser variable**

Diese Variablen werden direkt in Scheme definiert. Von ihrer direkten Benutzung durch den Benutzer wird streng abgeraten, weil ihre Semantikzuordnung sehr verwirrend sein kann.

Wenn der Wert einer derartigen Variable in einer `.ly`-Datei verändert wird, ist diese Änderung global, und wenn sie nicht explizit rückgängig gemacht wird, wird der neue Wert bis zum Ende der Datei gelten und dabei sowohl aufeinander folgende `\score`-Umbgebungen als auch externe Dateien, die mit `\include` geladen werden, beeinflussen. Das kann zu nicht gewollten Konsequenzen führen, und in komplizierteren Projekten kann es sehr schwer sein, die immer wieder auftretenden Fehler zu beheben.

LilyPond benutzt folgende Parser-Variablen:

- `afterGraceFraction`
- `musicQuotes`
- `mode`
- `output-count`
- `output-suffix`
- `partCombineListener`
- `pitchnames`
- `toplevel-bookparts`
- `toplevel-scores`
- `showLastLength`
- `showFirstLength`

**prob**


**simple closure**

Siehe `closure`, Seite 708.

**smob**

Smobs sind SceMe-OBJekte, Teile des Mechanismus von Guile, um C- und C++-Objekte in Scheme-Code zu exportieren. In LilyPond werden Smobs von C++-Objekten mithilfe von Makros...
erstellt. Es gibt zwei Arten von Smob-Objekten: einfache Smobs, die da sind für einfach immutable Objekte wie Nummern, und komplexe Smobs, benutzt für Objekte mit einer Identität. Wenn Sie auf die LilyPond-Quellen zurückgreifen können, findet sich mehr Information hierzu in lily/includes/smob.hh.

**stencil**

Eine Einheit der stencil-Klasse enthält die Information, die benötigt wird um ein typographisches Objekt zu setzen. Es handelt sich um einen sehr einfachen Smob, der eine begrenzende Box enthält, welche die vertikale und horizontale Ausdehnung des Objekt beschreibt, und einen Scheme-Ausdruck, der das Objekt ausgibt, nachdem es ausgewertet wurde. Stencil-Objekte können kombiniert werden, um komplexere Stencil zu bilden, die aus einem Baum von Scheme- Ausdrücken des Typ Stencil bestehen.

Die stencil-Eigenschaft, die einen Grob mit seinem Stencil verbindet, ist in der grob-interface-Schnittstelle definiert.

**Siehe auch**

Referenz der Interna: Abschnitt “grob-interface” in Referenz der Interna.

### A.16 Alle Kontexteigenschaften

- **accidentalGrouping (symbol)**
  - If set to 'voice, accidentals on the same note in different octaves may be horizontally staggered if in different voices.

- **additionalBassStrings (list)**
  - The additional tablature bass-strings, which will not get a separate line in TabStaff.
  - It is a list of the pitches of each string (starting with the lowest numbered one).

- **additionalPitchPrefix (string)**
  - Text with which to prefix additional pitches within a chord name.

- **aDueText (markup)**
  - Text to print at a unisono passage.

- **alignAboveContext (string)**
  - Where to insert newly created context in vertical alignment.

- **alignBassFigureAccidentals (boolean)**
  - If true, then the accidentals are aligned in bass figure context.

- **alignBelowContext (string)**
  - Where to insert newly created context in vertical alignment.

- **alternativeNumberingStyle (symbol)**
  - The style of an alternative's bar numbers. Can be numbers for going back to the same number or numbers-with-letters for going back to the same number with letter suffixes. No setting will not go back in measure-number time.

- **alternativeRestores (symbol list)**
  - Timing variables that are restored to their value at the start of the first alternative in subsequent alternatives.

- **associatedVoice (string)**
  - Name of the context (see associatedVoiceType for its type, usually Voice) that has the melody for this Lyrics line.

- **associatedVoiceType (symbol)**
  - Type of the context 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 Referenz der Interna then all staves share accidentals, and if context is Abschnitt “Staff” in Referenz der Interna 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:

caption 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.

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 false then bar lines will not be printed automatically; they must be explicitly created with a \bar command. Unlike the \cadenzaOn keyword, measures are still counted. Bar line 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.

barNumberFormatter (procedure)

A procedure that takes a bar number, measure position, and alternative number and returns a markup of the bar number to print.

barNumberVisibility (procedure)

A procedure that takes a bar number and a measure position and returns whether the corresponding bar number should be printed. Note that the actual print-out of bar numbers is controlled with the break-visibility property.
The following procedures are predefined:

**all-bar-numbers-visible**
Enable bar numbers for all bars, including the first one and broken bars (which get bar numbers in parentheses).

**first-bar-number-invisible**
Enable bar numbers for all bars (including broken bars) except the first one. If the first bar is broken, it doesn’t get a bar number either.

**first-bar-number-invisible-save-broken-bars**
Enable bar numbers for all bars (including broken bars) except the first one. A broken first bar gets a bar number.

**first-bar-number-invisible-and-no-parenthesized-bar-numbers**
Enable bar numbers for all bars except the first bar and broken bars. This is the default.

**(every-nth-bar-number-visible n)**
Assuming \(n\) is value 2, for example, this enables bar numbers for bars 2, 4, 6, etc.

**(modulo-bar-number-visible n m)**
If bar numbers 1, 4, 7, etc., should be enabled, \(n\) (the modulo) must be set to 3 and \(m\) (the division remainder) to 1.

**baseMoment (moment)**
Smallest unit of time that will stand on its own as a subdivided section.

**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.

**beamExceptions (list)**
An alist of exceptions to autobeam rules that normally end on beats.

**beamHalfMeasure (boolean)**
Whether to allow a beam to begin halfway through the measure in triple time, which could look like 6/8.

**beatStructure (list)**
List of baseMoments that are combined to make beats.

**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.

**chordNameLowercaseMinor (boolean)**
Downcase roots of minor chords?
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.

clefPosition (number)
Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

clefTransposition (integer)
Add this much extra transposition. Values of 7 and -7 are common.

clefTranspositionFormatter (procedure)
A procedure that takes the Transposition number as a string and the style as a symbol and returns a markup.

clefTranspositionStyle (symbol)
Determines the way the ClefModifier grob is displayed. Possible values are 'default', 'parenthesized' and 'bracketed'.

completionBusy (boolean)
Whether a completion-note head is playing.

completionFactor (an exact rational or procedure)
When Completion_heads_engraver and Completion_rest_engraver need to split a note or rest with a scaled duration, such as c2*3, this specifies the scale factor to use for the newly-split notes and rests created by the engraver.
If #f, the completion engraver uses the scale-factor of each duration being split.
If set to a callback procedure, that procedure is called with the context of the completion engraver, and the duration to be split.

completionUnit (moment)
Sub-bar unit of completion.

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' and 'text'. If unset, a hairpin crescendo is used.
crescendoText (markup)
The text to print at start of non-hairpin crescendo, i.e., ‘cresc.’.

cueClefGlyph (string)
Name of the symbol within the music font.

cueClefPosition (number)
Where should the center of the clef symbol go, measured in half staff spaces from
the center of the staff.

cueClefTransposition (integer)
Add this much extra transposition. Values of 7 and -7 are common.

cueClefTranspositionFormatter (procedure)
A procedure that takes the Transposition number as a string and the style as a
symbol and returns a markup.

cueClefTranspositionStyle (symbol)
Determines the way the ClefModifier grob is displayed. Possible values are ‘default’,
‘parenthesized’ and ‘bracketed’.

currentBarNumber (integer)
Contains the current bar number. This property is incremented at every bar line.

decrescendoSpanner (symbol)
The type of spanner to be used for decrescendi. Available values are ‘hairpin’ and
‘text’. If unset, a hairpin decrescendo is used.

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 Referenz der Interna at
Abschnitt “Score” in Referenz der Interna level.

defaultStrings (list)
A list of strings to use in calculating frets for tablatures and fretboards if no strings
are provided in the notes for the current moment.

doubleRepeatSegnoType (string)
Set the default bar line for the combinations double repeat with segno. Default is
‘:\.S.:’.

doubleRepeatType (string)
Set the default bar line for double repeats.

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’,
‘agostini-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.
endRepeatSegnoType (string)
Set the default bar line for the combinations ending of repeat with segno. Default
is ‘:|.S’.

endRepeatType (string)
Set the default bar line for the ending of repeats.

explicitClefVisibility (vector)
‘break-visibility’ function for clef changes.

explicitCueClefVisibility (vector)
‘break-visibility’ function for cue 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 that reduce the effect
of a previous alteration.

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.

fretLabels (list)
A list of strings or Scheme-formatted markups containing, in the correct order, the
labels to be used for lettered frets in tablature.
glissandoMap (list)
   A map in the form of '((source1 . target1) (source2 . target2) (sourceN . targetN))
   showing the glissandi to be drawn for note columns. The value '()' will default to
   '((0 . 0) (1 . 1) (n . n)), where n is the minimal number of note-heads in the two
   note columns between which the glissandi occur.

gridInterval (moment)
   Interval for which to generate GridPoints.

handleNegativeFrets (symbol)
   How the automatic fret calculator should handle calculated negative frets. Values
   include 'ignore, to leave them out of the diagram completely, 'include, to include
   them as calculated, and 'recalculate, to ignore the specified string and find a
   string where they will fit with a positive fret number.

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 Referenz der Interna line.

implicitBassFigures (list)
   A list of bass figures that are not printed as numbers, but only as extender lines.

includeGraceNotes (boolean)
   Do not ignore grace notes for Abschnitt “Lyrics” in Referenz der Interna.

initialTimeSignatureVisibility (vector)
   Break visibility for the initial 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 instrumentName property labels the staff in
   the first system, and the shortInstrumentName property labels following lines.

instrumentTransposition (pitch)
   Define the transposition of the instrument. Its value is the pitch that sounds when
   the instrument plays written 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).

keyAlterations (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. keyAlterations = #'(6 . ,FLAT)).

lyricMelismaAlignment (number)
Alignment to use for a melisma syllable.

magnifyStaffValue (positive number)
The most recent value set with \magnifyStaff.

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 two arguments: a TempoChangeEvent 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.

middleCCuePosition (number)
The position of the middle C, as determined only by the clef of the cue notes. This can be calculated by looking at cueClefPosition and cueClefGlyph.

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.

midiBalance (number)
Stereo balance for the MIDI channel associated with the current context. Ranges
from -1 to 1, where the values -1 (#LEFT), 0 (#CENTER) and 1 (#RIGHT) correspond
to leftmost emphasis, center balance, and rightmost emphasis, respectively.

midiChannelMapping (symbol)
How to map MIDI channels: per staff (default), instrument or voice.

midiChorusLevel (number)
Chorus effect level for the MIDI channel associated with the current context. Ranges
from 0 to 1 (0=off, 1=full effect).

midiExpression (number)
Expression control for the MIDI channel associated with the current context. Ranges
from 0 to 1 (0=off, 1=full effect).

midiInstrument (string)
Name of the MIDI instrument to use.

midiMaximumVolume (number)
Analogous to midiMinimumVolume.

midiMergeUnisons (boolean)
If true, output only one MIDI note-on event when notes with the same pitch, in the
same MIDI-file track, overlap.

midiMinimumVolume (number)
Set the minimum loudness for MIDI. Ranges from 0 to 1.

midiPanPosition (number)
Pan position for the MIDI channel associated with the current context. Ranges
from -1 to 1, where the values -1 (#LEFT), 0 (#CENTER) and 1 (#RIGHT) correspond
to hard left, center, and hard right, respectively.

midiReverbLevel (number)
Reverb effect level for the MIDI channel associated with the current context. Ranges
from 0 to 1 (0=off, 1=full effect).

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.

minorChordModifier (markup)
Markup displayed following the root for a minor chord

noChordSymbol (markup)
Markup to be displayed for rests in a ChordNames context.

noteToFretFunction (procedure)
Convert list of notes and list of defined strings to full list of strings and fret numbers.
Parameters: The context, a list of note events, a list of tabstring events, and the
fretboard grob if a fretboard is desired.
nullAccidentals (boolean)
   The Accidental_engraver generates no accidentals for notes in contexts where this
   is set. In addition to suppressing the printed accidental, this option removes any effect
   the note would have had on accidentals in other voices.

ottavation (markup)
   If set, the text for an ottava spanner. Changing this creates a new text spanner.

output (music output)
   The output produced by a score-level translator during music interpretation.

partCombineForced (symbol)
   Override for the partcombine decision. Can be apart, chords, unisono, solo1, or
   solo2.

partCombineTextsOnNote (boolean)
   Print part-combine texts only on the next note rather than immediately on rests or
   skips.

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 propor-
   tional notation.

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.
repeatCountVisibility (procedure)
   A procedure taking as arguments an integer and context, returning whether the corresponding percent repeat number should be printed when countPercentRepeats is set.

restCompletionBusy (boolean)
   Signal whether a completion-rest is active.

restNumberThreshold (number)
   If a multimeasure rest has more measures than this, a number is printed.

restrainOpenStrings (boolean)
   Exclude open strings from the automatic fret calculator.

searchForVoice (boolean)
   Signal whether a search should be made of all contexts in the context hierarchy for a voice to provide rhythms for the lyrics.

segnoType (string)
   Set the default bar line for a requested segno. Default is 'S'.

shapeNoteStyles (vector)
   Vector of symbols, listing style for each note head relative to the tonic (qv.) of the scale.

shortInstrumentName (markup)
   See instrumentName.

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.

slashChordSeparator (markup)
   The markup object used to separate a chord name from its root note in case of inversions or slash chords.

soloIIText (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 Referenz der Interna.
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.

startRepeatSegnoType (string)
   Set the default bar line for the combinations beginning of repeat with segno. Default is ‘S.|:’.

startRepeatType (string)
   Set the default bar line for the beginning of repeats.

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.

strictBeatBeaming (boolean)
   Should partial beams reflect the beat structure even if it causes flags to hang out?

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 pitches of each string (starting with the lowest numbered one).

strokeFingerOrientations (list)
   See fingeringOrientations.

subdivideBeams (boolean)
   If set, multiple beams will be subdivided at baseMoment positions by only drawing one beam over the beat.

suggestAccidentals (boolean)
   If set, accidentals are typeset as cautionary suggestions over the note.

supportNonIntegerFret (boolean)
   If set in Score the TabStaff will print micro-tones as ‘2 1/2’.

suspendRestMerging (boolean)
   When using the Merge_rest_engraver do not merge rests when this is set to true.

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: context, string number and, fret number. It returns the text as a markup.

tabStaffLineLayoutFunction (procedure)
   A function determining the staff position of a tablature note head. Called with two arguments: the context and the string.
tempoHideNote (boolean)
  Hide the note = count in tempo marks.

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 (fraction, as pair)
  A pair of numbers, signifying the time signature. For example, '(4 . 4) is a 4/4
  time signature.

timeSignatureSettings (list)
  A nested alist of settings for time signatures. Contains elements for various time
  signatures. The element for each time signature contains entries for baseMoment,
  beatStructure, and beamExceptions.

timing (boolean)
  Keep administration of measure length, position, bar number, etc.? Switch off for
  cadenzas.

tonic (pitch)
  The tonic of the current scale.

topLevelAlignment (boolean)
  If true, the Vertical_align_engraver will create a VerticalAlignment; otherwise, it
  will create a StaffGrouper.

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 \}
  \}

useBassFigureExtenders (boolean)
  Whether to use extender lines for repeated bass figures.

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 `scm/bar-line.scm`.

### A.17 Eigenschaften des Layouts

**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.

**annotation-balloon** (boolean)  
Print the balloon around an annotation.

**annotation-line** (boolean)  
Print the line from an annotation to the grob that it annotates.

**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 and it is larger than this number, make a kneed beam.

**automatically-numbered** (boolean)  
If set, footnotes are automatically numbered.

**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-scripts** (boolean)  
If set, a tuplet bracket avoids the scripts associated with the note heads it encompasses.
avoid-slur (symbol)
Method of handling slur collisions. Choices are inside, outside, around, and ignore. inside adjusts the slur if needed to keep the grob inside the slur. outside moves the grob vertically to the outside of the slur. around moves the grob vertically to the outside of the slur only if there is a collision. ignore does not move either. In grobs whose notational significance depends on vertical position (such as accidentals, clefs, etc.), outside and around behave like ignore.

axes (list) List of axis numbers. In the case of alignment grobs, this should contain only one number.

bar-extent (pair of numbers)
The Y-extent of the actual bar line. This may differ from Y-extent because it does not include the dots in a repeat 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.

beamlet-default-length (pair)
A pair of numbers. The first number specifies the default length of a beamlet that sticks out of the left hand side of this stem; the second number specifies the default length of the beamlet to the right. The actual length of a beamlet is determined by taking either the default length or the length specified by beamlet-max-length-proportion, whichever is smaller.

beamlet-max-length-proportion (pair)
The maximum length of a beamlet, as a proportion of the distance between two adjacent stems.

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.
Anhang A: Notationsübersicht

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 breakable item 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)
This is a vector of 3 lists: #(end-of-line unbroken start-of-line). Each list contains break-align symbols that specify an order of breakable items (see Abschnitt “break-alignment-interface” in Referenz der Interna).
For example, this places time signatures before clefs:

```
\override Score.BreakAlignment.break-align-orders = 
  #(make-vector 3 'left-edge
    cue-end-clef
    ambitus
    breathing-sign
    time-signature
    clef
    cue-clef
    staff-bar
    key-cancellation
    key-signature
    custos))
```

break-align-symbol (symbol)
This key is used for aligning, ordering, and spacing breakable items. See Abschnitt “break-alignment-interface” in Referenz der Interna.

break-align-symbols (list)
A list of break-align symbols that determines which breakable items 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). Choices are listed in Abschnitt “break-alignment-interface” in Referenz der Interna.

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.

broken-bound-padding (number)
The amount of padding to insert when a spanner is broken at a line break.
chord-dots-limit (integer)
   Limits the column of dots on each chord to the height of the chord plus chord-dots-limit staff-positions.

circled-tip (boolean)
   Put a circle at start/end of hairpins (al/del niente).

clef-alignments (list)
   An alist of parent-alignments that should be used for clef modifiers with various clefs

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.

collision-interfaces (list)
   A list of interfaces for which automatic beam-collision resolution is run.

collision-voice-only (boolean)
   Does automatic beam collision apply only to the voice in which the beam was created?

color (color)
   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 of number pairs)
   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.

count-from (integer)
   The first measure in a measure count receives this number. The following measures are numbered in increments from this initial value.

damping (number)
   Amount of beam slope damping.

dash-definition (pair)
   List of dash-elements defining the dash structure. Each dash-element has a starting t value, an ending t-value, a dash-fraction, and a dash-period.

dash-fraction (number)
   Size of the dashes, relative to dash-period. Should be between 0.1 and 1.0 (continuous line). If set to 0.0, a dotted line is produced

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.

default-staff-staff-spacing (list)
The settings to use for staff-staff-spacing when it is unset, for ungrouped staves and for grouped staves that do not have the relevant StaffGrouper property set (staff-staff-spacing or staffgroup-staff-spacing).

details (list)
A list of parameters for detailed grob behavior. More information on the allowed parameters for a grob can be found by looking at the top of the Internals Reference page for each interface having a details property.
digit-names (vector)
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-negative-kern (number)
The space to remove between a dot and a slash in percent repeat glyphs. Larger values bring the two elements closer together.
dot-placement-list (list)
List consisting of (description string-number fret-number finger-number) entries used to define fret diagrams.
double-stem-separation (number)
The distance between the two stems of a half note in tablature when using \tabFullNotation, not counting the width of the stems themselves, expressed as a multiple of the default height of a staff-space in the traditional five-line staff.
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-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-count (number)
The number of tremolo beams.

flag-style (symbol)
The style of the flag to be used with MetronomeMark. Available are 'modern-straight-flag', 'old-straight-flag', flat-flag, mensural and 'default'.

flat-positions (list)
Flats in key signatures are placed within the specified ranges of staff-positions. The general form is a list of pairs, with one pair for each type of clef, in order of the staff-position at which each clef places C: (alto treble tenor soprano baritone mezzosoprano bass). If the list contains a single element it applies for all clefs. A single number in place of a pair sets accidentals within the octave ending at that staff-position.

font-encoding (symbol)
The font encoding is the broadest category for selecting a font. Currently, only Lilypond's system fonts (Emmentaler) are using this property. Available values are fetaMusic (Emmentaler), fetaBraces, fetaText (Emmentaler).

font-family (symbol)
The font family is the broadest category for selecting text fonts. Options include: sans, roman.

font-features (list)
Opentype features.

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. If the context property fontSize is set, its value is added to this before the glyph is printed. Fractional values are allowed.

footnote (boolean)
Should this be a footnote or in-note?
footnote-music (music)
Music creating a footnote.

footnote-text (markup)
A footnote for the grob.

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 Referenz der Interna.

forced-spacing (number)
Spacing forced between grobs, used in various ligature engravers.

fraction (fraction, as pair)
Numerator and denominator of a time signature object.

defrench-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.
• capo-thickness – Thickness of capo indicator, in multiples of fret-space. Default value 0.5.
• 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-position – Location of dot in fret space. Default 0.6 for dots without labels, 0.95-dot-radius for dots with labels.
• 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-distance – Multiplier to adjust the distance between frets. Default 1.0.
• fret-label-custom-format – The format string to be used label the lowest fret number, when number-type equals to custom. Default "-a".
• fret-label-font-mag – The magnification of the font used to label the lowest fret number. Default 0.5.
• fret-label-vertical-offset – The offset of the fret label from the center of the fret in direction parallel to strings. Default 0.
• fret-label-horizontal-offset – The offset of the fret label from the center of the fret in direction orthogonal to strings. Default 0.
• paren-padding – The padding for the parenthesis. Default 0.05.
• 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, arabic and custom. In the later case, the format string is supplied by the fret_LABEL_CUSTOM_FORMAT property. Default roman_LOWER.

• **open-string** – Character string to be used to indicate open string. Default "o".

• **orientation** – Orientation of fret-diagram. Options include normal, landscape, and opposing-landscape. Default normal.

• **string-count** – The number of strings. Default 6.

• **string-distance** – Multiplier to adjust the distance between strings. Default 1.0.

• **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 for normal orientation, 0.5 for landscape and opposing-landscape.

• **string-thickness-factor** – Factor for changing thickness of each string in the fret diagram. Thickness of string $k$ is given by $\text{thickness} \times (1 + \text{string-thickness-factor})^k$. Default 0.

• **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-measure-extra-space** (number)
Extra space that is allocated at the beginning of a measure with only one note. This property is read from the NonMusicalPaperColumn that begins the measure.

**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.

**glissando-skip** (boolean)
Should this NoteHead be skipped by glissandi?

**glyph** (string)
A string determining what 'style' of glyph is typeset. Valid choices depend on the function that is reading this property.

In combination with (span) bar lines, it is a string resembling the bar line appearance in ASCII form.
glyph-name (string)
The glyph name within the font.
In the context of (span) bar lines, glyph-name represents a processed form of glyph, where decisions about line breaking etc. are already taken.

glyph-name-alist (list)
An alist of key-string pairs.

graphical (boolean)
Display in graphical (vs. text) form.

grow-direction (direction)
Crescendo or decrescendo?

hair-thickness (number)
Thickness of the thin line in a bar line, expressed as a multiple of the default staff-line thickness (i.e. the visual output is not influenced by changes to Staff.StaffSymbol.thickness).

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.

hide-tied-accidental-after-break (boolean)
If set, an accidental that appears on a tied note after a line break will not be displayed.

horizon-padding (number)
The amount to pad the axis along which a Skyline is built for the side-position-interface.
horizontal-shift (integer)
   An integer that identifies ranking of NoteColumns for horizontal shifting. This is
   used by Abschnitt “note-collision-interface” in Referenz der Interna.

horizontal-skylines (pair of skylines)
   Two skylines, one to the left and one to the right of this grob.

id (string)
   An id string for the grob.

ignore-ambitus (boolean)
   If set, don’t consider this notehead for ambitus calculation.

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-inside-line (boolean)
   If set, this column cannot have objects sticking into the margin.

kern (dimension, in staff space)
   The space between individual elements in any compound bar line, expressed as a
   multiple of the default staff-line thickness (i.e. the visual output is not influenced
   by changes to Staff.StaffSymbol.thickness).

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.

knee-to-beam (boolean)
   Determines whether a tuplet number will be positioned next to a kneed beam.

labels (list)
   List of labels (symbols) placed on a column.

layer (integer)
   An integer which determines the order of printing objects. Objects with the lowest
   value of layer are drawn first, then objects with progressively higher values are
   drawn, so objects with higher values overwrite objects with lower values. By default
   most objects are assigned a layer value of 1.

ledger-extra (dimension, in staff space)
   Extra distance from staff line to draw ledger lines for.

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.
ledger-positions (list)
Vertical positions of ledger lines. When set on a StaffSymbol grob it defines a
repeating pattern of ledger lines and any parenthesized groups will always be shown
together.

ledger-positions-function (any type)
A quoted Scheme procedure that takes a StaffSymbol grob and the vertical position
of a note head as arguments and returns a list of ledger line positions.

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 lyric extender).

length (dimension, in staff space)
User override for the stem length of unbeam stems (each unit represents half a
staff-space).

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)
For slurs and ties, this is the diameter of the virtual „pen“ that draws the two arcs
of the curve’s outline, which intersect at the endpoints. This property is expressed
as a multiple of the current staff-line thickness (i.e. the visual output is influenced
by changes to Staff.StaffSymbol.thickness).

long-text (markup)
Text markup. See Abschnitt “Formatting text” in Notationsreferenz.

max-beam-connect (integer)
Maximum number of beams to connect to beams from this stem. Further beams are
typeset as beamlets.

max-symbol-separation (number)
The maximum distance between symbols making up a church rest.

maximum-gap (number)
Maximum value allowed for gap property.

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 Referenz der Interna.
merge-differently-headed only applies to opposing stem directions (i.e., voice 1 & 2).

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-after-break (dimension, in staff space)
If set, try to make a broken spanner starting a line this long. This requires an appropriate callback for the springs-and-rods property. If added to a Tie, this sets the minimum distance to the notehead.

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).

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.

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 (graphical (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-break-align-symbols (list)
A list of symbols that determine which NON-break-aligned interfaces to align this to.

non-default (boolean)
Set for manually specified clefs and keys.

non-musical (boolean)
True if the grob belongs to a NonMusicalPaperColumn.

nonstaff-nonstaff-spacing (list)
The spacing alist controlling the distance between the current non-staff line and the next non-staff line in the direction of staff-affinity, if both are on the same side of the related staff, and staff-affinity is either UP or DOWN. See staff-staff-spacing for a description of the alist structure.

nonstaff-relatedstaff-spacing (list)
The spacing alist controlling the distance between the current non-staff line and the nearest staff in the direction of staff-affinity, if there are no non-staff lines between the two, and staff-affinity is either UP or DOWN. If staff-affinity is CENTER, then nonstaff-relatedstaff-spacing is used for the nearest staves on both sides, even if other non-staff lines appear between the current one and either of the staves. See staff-staff-spacing for a description of the alist structure.

nonstaff-unrelatedstaff-spacing (list)
The spacing alist controlling the distance between the current non-staff line and the nearest staff in the opposite direction from staff-affinity, if there are no other non-staff lines between the two, and staff-affinity is either UP or DOWN. See staff-staff-spacing for a description of the alist structure.

normalized-endpoints (pair)
Represents left and right placement over the total spanner, where the width of the spanner is normalized between 0 and 1.

note-collision-threshold (dimension, in staff space)
Simultaneous notes that are this close or closer in units of staff-space will be identified as vertically colliding. Used by Stem grobs for notes in the same voice, and NoteCollision grobs for notes in different voices. Default value 1.

note-names (vector)
Vector of strings containing names for easy-notation note heads.

number-type (symbol)
Numbering style. Choices include roman-lower, roman-upper and arabic.

output-attributes (list)
An alist of attributes for the grob, to be included in output files. When the SVG typesetting backend is used, the attributes are assigned to a group (&lt;g&gt;) containing all of the stencils that comprise a given grob. For example, '&(id . 123) (class . foo) (data-whatever . ,bar\))' will produce &lt;g id=&quot;123\ class=&quot;foo\ data-whatever=.bar\&quot;\&apos;\&apos; ... &lt;/g&gt;'. In the Postscript backend, where there is no way to group items, the setting of the output-attributes property will have no effect.

outside-staff-horizontal-padding (number)
By default, an outside-staff-object can be placed so that it is 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 grobs when spacing according to outside-staff-priority. Two grobs with different outside-staff-padding values have the larger value of padding between them.

outside-staff-placement-directive (symbol)
One of four directives telling how outside staff objects should be placed.

- left-to-right-greedy – Place each successive grob from left to right.
- left-to-right-polite – Place a grob from left to right only if it does not potentially overlap with another grob that has been placed on a pass through a grob array. If there is overlap, do another pass to determine placement.
- right-to-left-greedy – Same as left-to-right-greedy, but from right to left.
- right-to-left-polite – Same as left-to-right-polite, but from right to left.

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.

padding-pairs (list)
An alist mapping (name . name) to distances.

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.

parent-alignment-X (number)
Specify on which point of the parent the object is aligned. The value -1 means aligned on parent’s left edge, 0 on center, and 1 right edge, in X direction. Other numerical values may also be specified - the unit is half the parent’s width. If unset, the value from self-alignment-X property will be used.

parent-alignment-Y (number)
Like parent-alignment-X but for the Y axis.
parenthesis-friends (list)
A list of Grob types, as symbols. When parentheses enclose a Grob that has `parenthesis-friends, the parentheses widen to include any child Grobs with type among `parenthesis-friends.

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.

protrusion (number)
In an arpeggio bracket, the length of the horizontal edges.

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 an orchestral score?

remove-layer (index or symbol)
When set as a positive integer, the Keep_alive_together_ engraver removes all VerticalAxisGroup grobs with a remove-layer larger than the smallest retained remove-layer. Set to #f to make a layer independent of the Keep_alive_together_ engraver. Set to '(), the layer does not participate in the layering decisions. The property can also be set as a symbol for common behaviors: #'any to keep the layer alive with any other layer in the group; #'above or #'below to keep the layer alive with the context immediately before or after it, respectively.

replacement-alist (list)
Alist of strings. The key is a string of the pattern to be replaced. The value is a string of what should be displayed. Useful for ligatures.

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.
round-up-exceptions (list)
A list of pairs where car is the numerator and cdr the denominator of a moment. Each pair in this list means that the multi-measure rests of the corresponding length will be rounded up to the longer rest. See round-up-to-longer-rest.

round-up-to-longer-rest (boolean)
Displays the longer multi-measure rest when the length of a measure is between two values of usable-duration-logs. For example, displays a breve instead of a whole in a 3/2 measure.

rounded (boolean)
Decide whether lines should be drawn rounded or not.

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 key for determining the order of scripts in a stack, by being added to the position of the script in the user input, the sum being the overall priority. Smaller means closer to the head.

segno-kern (number)
The space between the two thin lines of the segno bar line symbol, expressed as a multiple of the default staff-line thickness (i.e. the visual output is not influenced by changes to Staff.StaffSymbol.thickness).

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 - the unit is half the object width.

self-alignment-Y (number)
Like self-alignment-X but for the Y axis.

shape (symbol)
This setting determines what shape a grob has. Valid choices depend on the stencil callback reading this property.

sharp-positions (list)
Sharps in key signatures are placed within the specified ranges of staff-positions. The general form is a list of pairs, with one pair for each type of clef, in order of the staff-position at which each clef places C: (alto treble tenor soprano baritone mezzosoprano bass). If the list contains a single element it applies for all clefs. A single number in place of a pair sets accidentals within the octave ending at that staff-position.

shorten-pair (pair of numbers)
The lengths to shorten on both sides a hairpin or text-spanner such as a pedal bracket. Positive values shorten the hairpin or text-spanner, while negative values lengthen it.

shortest-duration-space (number)
Start with this multiple of spacing-increment space for the shortest duration. See also Abschnitt “spacing-spanner-interface” in Referenz der Interna.

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.

simple-Y (boolean)
Should the Y placement of a spanner disregard changes in system heights?

size (number)
The ratio of the size of the object to its default size.

skip-quanting (boolean)
Should beam quanting be skipped?

skyline-horizontal-padding (number)
For determining the vertical distance between two staves, it is possible to have a configuration which would result in a tight interleaving of grobs from the top staff and the bottom staff. The larger this parameter is, the farther apart the staves are placed in such a configuration.

skyline-vertical-padding (number)
The amount by which the left and right skylines of a column are padded vertically, beyond the Y-extents and extra-spacing-heights of the constituent grobs in the column. Increase this to prevent interleaving of grobs from adjacent columns.

slash-negative-kern (number)
The space to remove between slashes in percent repeat glyphs. Larger values bring the two elements closer together.

slope (number)
The slope of this object.

slur-padding (number)
Extra distance between slur and script.

snap-radius (number)
The maximum distance between two objects that will cause them to snap to alignment along an axis.

space-alist (list)
An alist that specifies distances from this grob to other breakable items, using the format:

'((break-align-symbol . (spacing-style . space))
  (break-align-symbol . (spacing-style . space))
  ...)

Standard choices for break-align-symbol are listed in Abschnitt “break-alignment-interface” in Referenz der Interna. Additionally, three special break-align symbols available to space-alist are:

  first-note
  used when the grob is just left of the first note on a line

  next-note
  used when the grob is just left of any other note; if not set, the value of first-note gets used
right-edge
used when the grob is the last item on the line (only compatible with the extra-space spacing style)

Choices for spacing-style are:

extra-space
Put this much space between the two grobs. The space is stretchable when paired with first-note or next-note; otherwise it is fixed.

minimum-space
Put at least this much space between the left sides of both grobs, without allowing them to collide. The space is stretchable when paired with first-note or next-note; otherwise it is fixed. Not compatible with right-edge.

fixed-space
Only compatible with first-note and next-note. Put this much fixed space between the grob and the note.

minimum-fixed-space
Only compatible with first-note and next-note. Put at least this much fixed space between the left side of the grob and the left side of the note, without allowing them to collide.

semi-fixed-space
Only compatible with first-note and next-note. Put this much space between the grob and the note, such that half of the space is fixed and half is stretchable.

Rules for this spacing are much more complicated than this. See [Wanske] page 126–134, [Ross] page 143–147.

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 (dimension, in staff space)
The unit of length for note-spacing. Typically, the width of a note head. See also Abschnitt “spacing-spanner-interface” in Referenz der Interna.

spacing-pair (pair)
A pair of alignment symbols which set an object’s spacing relative to its left and right BreakAlignments.

For example, a MultiMeasureRest will ignore prefatory items at its bounds (i.e., clefs, key signatures and time signatures) using the following override:

\override MultiMeasureRest.spacing-pair = #'(staff-bar . staff-bar)

spanner-id (index or symbol)
An identifier to distinguish concurrent spanners.

springs-and-rods (boolean)
Dummy variable for triggering spacing routines.
stacking-dir (direction)
Stack objects in which direction?

staff-affinity (direction)
The direction of the staff to use for spacing the current non-staff line. Choices are UP, DOWN, and CENTER. If CENTER, the non-staff line will be placed equidistant between the two nearest staves on either side, unless collisions or other spacing constraints prevent this. Setting staff-affinity for a staff causes it to be treated as a non-staff line. Setting staff-affinity to #f causes a non-staff line to be treated as a staff.

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.

staff-staff-spacing (list)
When applied to a staff-group’s StaffGrouper grob, this spacing alist controls the distance between consecutive staves within the staff-group. When applied to a staff’s VerticalAxisGroup grob, it controls the distance between the staff and the nearest staff below it in the same system, replacing any settings inherited from the StaffGrouper grob of the containing staff-group, if there is one. This property remains in effect even when non-staff lines appear between staves. The alist can contain the following keys:

- basic-distance – the vertical distance, measured in staff-spaces, between the reference points of the two items when no collisions would result, and no stretching or compressing is in effect.
- minimum-distance – the smallest allowable vertical distance, measured in staff-spaces, between the reference points of the two items, when compressing is in effect.
- padding – the minimum required amount of unobstructed vertical whitespace between the bounding boxes (or skylines) of the two items, measured in staff-spaces.
- stretchability – a unitless measure of the dimension’s relative propensity to stretch. If zero, the distance will not stretch (unless collisions would result).

staffgroup-staff-spacing (list)
The spacing alist controlling the distance between the last staff of the current staff-group and the staff just below it in the same system, even if one or more non-staff lines exist between the two staves. If the staff-staff-spacing property of the staff’s VerticalAxisGroup grob is set, that is used instead. See staff-staff-spacing for a description of the alist structure.

stem-attachment (pair of numbers)
An (x, y) pair where the stem attaches to the notehead.

stem-begin-position (number)
User override for the begin position of a stem.
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 be a stem over a rest?

stencil (stencil)
The symbol to print.

stencils (list)
Multiple stencils, used as intermediate value.

strict-grace-spacing (boolean)
If set, main notes are spaced normally, then grace notes are put left of the musical
columns for the main notes.

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 “Formatting text” in Notationsreferenz.

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)
Thickness of the thick line in a bar line, expressed as a multiple of the de-
fault staff-line thickness (i.e. the visual output is not influenced by changes to
Staff.StaffSymbol.thickness).

thickness (number)
For grobs made up of lines, this is the thickness of the line. For slurs and ties, this
is the distance between the two arcs of the curve’s outline at its thickest point, not
counting the diameter of the virtual „pen“ that draws the arcs. This property is
expressed as a multiple of the current staff-line thickness (i.e. the visual output is
influenced by changes to Staff.StaffSymbol.thickness).

tie-configuration (list)
List of (position . dir) pairs, indicating the desired tie configuration, where po-
sition 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.

toward-stem-shift (number)
Amount by which scripts are shifted toward the stem if their direction coincides
with the stem direction. 0.0 means centered on the note head (the default position
of most scripts); 1.0 means centered on the stem. Interpolated values are possible.
toward-stem-shift-in-column (number)
   Amount by which a script is shifted toward the stem if its direction coincides with the
   stem direction and it is associated with a ScriptColumn object. 0.0 means centered
   on the note head (the default position of most scripts); 1.0 means centered on the
   stem. Interpolated values are possible.

transparent (boolean)
   This makes the grob invisible.

uniform-stretching (boolean)
   If set, items stretch proportionally to their natural separation based on durations.
   This looks better in complex polyphonic patterns.

usable-duration-logs (list)
   List of duration-logs that can be used in typesetting the grob.

use-skylines (boolean)
   Should skylines be used for side positioning?

used (boolean)
   If set, this spacing column is kept in the spacing problem.

vertical-skylines (pair of skylines)
   Two skylines, one above and one below this grob.

voiced-position (number)
   The staff-position of a voiced Rest, negative if the rest has direction DOWN.

when (moment)
   Global time step associated with this column.

whiteout (boolean-or-number)
   If a number or true, the grob is printed over a white background to white-out
   underlying material, if the grob is visible. A number indicates how far the white
   background extends beyond the bounding box of the grob as a multiple of the staff-
   line thickness. The LyricHyphen grob uses a special implementation of whiteout:
   A positive number indicates how far the white background extends beyond the
   bounding box in multiples of line-thickness. The shape of the background is
   determined by whiteout-style. Usually #f by default.

whiteout-style (symbol)
   Determines the shape of the whiteout background. Available are 'outline,
   'rounded-box, and the default 'box. There is one exception: Use 'special for
   LyricHyphen.

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.

X-align-on-main-noteheads (boolean)
   If true, this grob will ignore suspended noteheads when aligning itself on NoteCol-
   umn.

X-extent (pair of numbers)
   Extent (size) in the X direction, measured in staff-space units, relative to object’s
   reference point.

X-offset (number)
   The horizontal amount that this object is moved relative to its X-parent.
X-positions (pair of numbers)
Pair of X staff coordinates of a spanner in the form $\text{left} . \text{right}$, where both $\text{left}$ and $\text{right}$ are in staff-space units of the current staff.

Y-extent (pair of numbers)
Extent (size) in the Y direction, measured in staff-space units, relative to object’s reference point.

Y-offset (number)
The vertical amount that this object is moved relative to its Y-parent.

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.

### A.18 Erhältliche Musikfunktionen

**absolute** [music] - music (music)
Make music absolute. This does not actually change the music itself but rather hides it from surrounding \relative and \fixed commands.

**acciaccatura** [music] - music (music)
Create an acciaccatura from the following music expression.

**accidentalStyle** [music] - style (symbol list)
Set accidental style to symbol list style in the form ‘piano-cautionary’. If style has a form like ‘Staff.piano-cautionary’, the settings are applied to that context. Otherwise, the context defaults to ‘Staff’, except for piano styles, which use ‘GrandStaff’ as a context.

**addChordShape** [void] - key-symbol (symbol) tuning (pair) shape-definition (string or pair)
Add chord shape shape-definition to the chord-shape-table hash with the key (cons key-symbol tuning).

**addInstrumentDefinition** [void] - name (string) lst (list)
Create instrument name with properties list.

**addQuote** [void] - name (string) music (music)
Define music as a quotable music expression named name.

**afterGrace** [music] - fraction [fraction, as pair] main (music) grace (music)
Create grace note(s) after a main music expression.

The musical position of the grace expression is after a given fraction of the main note’s duration has passed. If fraction is not specified as first argument, it is taken from afterGraceFraction which has a default value of 3/4.

**allowPageTurn** [music]
Allow a page turn. May be used at toplevel (ie between scores or markups), or inside a score.

**allowVoltaHook** [void] - bar (string)
Allow the volta bracket hook being drawn over bar line bar.

**alterBroken** [music] - property (key list or symbol) arg (list) item (key list or music)
Override property for pieces of broken spanner item with values arg. item may either be music in the form of a starting spanner event, or a symbol list in the form
‘Context.Grob’ or just ‘Grob’. If item is in the form of a spanner event, property may also have the form ‘Grob.property’ for specifying a directed tweak.

appendToTag [music] - tag (symbol) more (music) music (music)
    Append more to the elements of all music expressions in music that are tagged with tag.

applyContext [music] - proc (procedure)
    Modify context properties with Scheme procedure proc.

applyMusic [music] - func (procedure) music (music)
    Apply procedure func to music.

applyOutput [music] - target (symbol list or symbol) proc (procedure)
    Apply function proc to every layout object matched by target which takes the form Context or Context.Grob.

appoggiatura [music] - music (music)
    Create an appoggiatura from music

assertBeamQuant [music] - l (pair) r (pair)
    Testing function: check whether the beam quants l and r are correct

assertBeamSlope [music] - comp (procedure)
    Testing function: check whether the slope of the beam is the same as comp

    Make voices that switch between staves automatically. As an option the pitch where to switch staves may be specified. The clefs for the staves are optional as well. Setting clefs works only for implicitly instantiated staves.

balloonGrobText [music] - grob-name (symbol) offset (pair of numbers) text (markup)
    Attach text to grob-name at offset offset (use like \once)

balloonText [post event] - offset (pair of numbers) text (markup)
    Attach text at offset (use like \tweak)

bar [music] - type (string)
    Insert a bar line of type type

barNumberCheck [music] - n (integer)
    Print a warning if the current bar number is not n.

beamExceptions (any type) - music (music)
    Extract a value suitable for setting Timing.beamExceptions from the given pattern with explicit beams in music. A bar check | has to be used between bars of patterns in order to reset the timing.

bendAfter [post event] - delta (real number)
    Create a fall or doit of pitch interval delta.

bookOutputName [void] - newfilename (string)
    Direct output for the current book block to newfilename.

bookOutputSuffix [void] - newsuffix (string)
    Set the output filename suffix for the current book block to newsuffix.

breathe [music]
    Insert a breath mark.
**chordRepeats** [music] - event-types [list] music (music)

Walk through music putting the notes of the previous chord into repeat chords, as well as an optional list of event-types such as #'(string-number-event).

**clef** [music] - type (string)

Set the current clef to type.

**compoundMeter** [music] - args (pair)

Create compound time signatures. The argument is a Scheme list of lists. Each list describes one fraction, with the last entry being the denominator, while the first entries describe the summands in the numerator. If the time signature consists of just one fraction, the list can be given directly, i.e. not as a list containing a single list. For example, a time signature of (3+1)/8 + 2/4 would be created as \compoundMeter #'((3 1 8) (2 4)), and a time signature of (3+2)/8 as \compoundMeter #'((3 2 8)) or shorter \compoundMeter #'(3 2 8).

**compressMMRests** [music] - music (music)

Remove the empty bars created by multi-measure rests, leaving just the first bar containing the MM rest itself.

**crossStaff** [music] - notes (music)

Create cross-staff stems.

**cueClef** [music] - type (string)

Set the current cue clef to type.

**cueClefUnset** [music]

Unset the current cue clef.

**cueDuring** [music] - what (string) dir (direction) main-music (music)

Insert contents of quote what corresponding to main-music, in a CueVoice oriented by dir.

**cueDuringWithClef** [music] - what (string) dir (direction) clef (string) main-music (music)

Insert contents of quote what corresponding to main-music, in a CueVoice oriented by dir.

**deadNote** [music] - note (music)

Print note with a cross-shaped note head.

**defineBarLine** [void] - bar (string) glyph-list (list)

Define bar line settings for bar line bar. The list glyph-list must have three entries which define the appearance at the end of line, at the beginning of the next line, and the span bar, respectively.

**displayLilyMusic** [music] - port [output port] music (music)

Display the LilyPond input representation of music to port, defaulting to the console.

**displayMusic** [music] - port [output port] music (music)

Display the internal representation of music to port, defaulting to the console.

**displayScheme** (any type) - port [output port] expr (any type)

Display the internal representation of expr to port, defaulting to the console.

**endSpanners** [music] - music (music)

Terminate the next spanner prematurely after exactly one note without the need of a specific end spanner.

**eventChords** [music] - music (music)

Compatibility function wrapping EventChord around isolated rhythmic events occurring since version 2.15.28, after expanding repeat chords ‘q’.
featherDurations [music] - factor (moment) argument (music)
Adjust durations of music in argument by rational factor.

finger [post event] - finger (number or markup)
Apply finger as a fingering indication.

fixed [music] - pitch (pitch) music (music)
Use the octave of pitch as the default octave for music.

footnote [music] - mark [markup] offset (pair of numbers) footnote (markup) item (symbol list or music)
Make the markup footnote a footnote on item. The footnote is marked with a markup mark moved by offset with respect to the marked music.
If mark is not given or specified as \default, it is replaced by an automatically generated sequence number. If item is a symbol list of form ‘Grob’ or ‘Context.Grob’, then grobs of that type will be marked at the current time step in the given context (default Bottom).
If item is music, the music will get a footnote attached to a grob immediately attached to the event, like \tweak does. For attaching a footnote to an indirectly caused grob, write \single\footnote, use item to specify the grob, and follow it with the music to annotate.
Like with \tweak, if you use a footnote on a following post-event, the \footnote command itself needs to be attached to the preceding note or rest as a post-event with -.

grace [music] - music (music)
Insert music as grace notes.

grobdescriptions (any type) - descriptions (list)
Create a context modification from descriptions, a list in the format of all-grob-descriptions.

harmonicByFret [music] - fret (number) music (music)
Convert music into mixed harmonics; the resulting notes resemble harmonics played on a fretted instrument by touching the strings at fret.

harmonicByRatio [music] - ratio (number) music (music)
Convert music into mixed harmonics; the resulting notes resemble harmonics played on a fretted instrument by touching the strings at the point given through ratio.

harmonicNote [music] - note (music)
Print note with a diamond-shaped note head.

harmonicsOn [music]
Set the default note head style to a diamond-shaped style.

hide [music] - item (symbol list or music)
Set item’s ‘transparent’ property to #t, making it invisible while still retaining its dimensions.
If item is a symbol list of form GrobName or Context.GrobName, the result is an override for the grob name specified by it. If item is a music expression, the result is the same music expression with an appropriate tweak applied to it.

incipit [music] - incipit-music (music)
Output incipit-music before the main staff as an indication of its appearance in the original music.
inherit-acceptability [void] - to (symbol) from (symbol)
When used in an output definition, will modify all context definitions such that
context to is accepted as a child by all contexts that also accept from.

inStaffSegno [music]
Put the segno variant 'varsegno' at this position into the staff, compatible with the
repeat command.

instrumentSwitch [music] - name (string)
Switch instrument to name, which must be predefined with
\addInstrumentDefinition.

inversion [music] - around (pitch) to (pitch) music (music)
Invert music about around and transpose from around to to.

keepWithTag [music] - tags (symbol list or symbol) music (music)
Include only elements of music that are tagged with one of the tags in tags. tags
may be either a single symbol or a list of symbols.
Each tag may be declared as a member of at most one tag group (defined with\tagGroup). If none of a music element’s tags share a tag group with one of the
specified tags, the element is retained.

key [music] - tonic [pitch] pitch-alist [list]
Set key to tonic and scale pitch-alist. If both are null, just generate KeyChangeEvent.

killCues [music] - music (music)
Remove cue notes from music.

label [music] - label (symbol)
Create label as a bookmarking label.

language [void] - language (string)
Set note names for language language.

languageRestore [void]
Restore a previously-saved pitchnames alist.

languageSaveAndChange [void] - language (string)
Store the previous pitchnames alist, and set a new one.

magnifyMusic [music] - mag (positive number) music (music)
Magnify the notation of music without changing the staff-size, using mag as a size
factor. Stems, beams, slurs, ties, and horizontal spacing are adjusted automatically.

magnifyStaff [music] - mag (positive number)
Change the size of the staff, adjusting notation size and horizontal spacing automatical-ly, using mag as a size factor.

makeClusters [music] - arg (music)
Display chords in arg as clusters.

makeDefaultStringTuning [void] - symbol (symbol) pitches (list)
This defines a string tuning symbol via a list of pitches. The symbol also gets
registered in defaultStringTunings for documentation purposes.

mark [music] - label [number or markup]
Make the music for the \mark command.

markupMap [music] - path (symbol list or symbol) markupfun (markup-function) music (music)
This applies the given markup function markupfun to all markup music properties
matching path in music.
For example,
\new Voice { g'2 c'' }
\addlyrics {
  \markupMap LyricEvent.text
  \markup \with-color #red \etc
  \{ Oh yes! \}
}

\textbf{modalInversion} [music] - around (pitch) to (pitch) scale (music) music (music)
Invert music about around using scale and transpose from around to to.

\textbf{modalTranspose} [music] - from (pitch) to (pitch) scale (music) music (music)
Transpose music from pitch from to pitch to using scale.

\textbf{musicMap} [music] - proc (procedure) mus (music)
Apply proc to mus and all of the music it contains.

\textbf{noPageBreak} [music]
Forbid a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.

\textbf{noPageTurn} [music]
Forbid a page turn. May be used at toplevel (i.e., between scores or markups), or inside a score.

\textbf{octaveCheck} [music] - pitch (pitch)
Octave check.

\textbf{offset} [music] - property (symbol list or symbol) offsets (any type) item (key list or music)
Offset the default value of property of item by offsets. If item is a string, the result is \override for the specified grob type. If item is a music expression, the result is the same music expression with an appropriate tweak applied.

\textbf{omit} [music] - item (symbol list or music)
Set item’s `stencil` property to #f, effectively omitting it without taking up space. If item is a symbol list of form GrobName or Context.GrobName, the result is an override for the grob name specified by it. If item is a music expression, the result is the same music expression with an appropriate tweak applied to it.

\textbf{once} [music] - music (music)
Set once to #t on all layout instruction events in music. This will complain about music with an actual duration. As a special exception, if music contains ‘tweaks’ it will be silently ignored in order to allow for \once \propertyTweak to work as both one-time override and proper tweak.

\textbf{ottava} [music] - octave (integer)
Set the octavation.

\textbf{overrideProperty} [music] - grob-property-path (list of indexes or symbols) value (any type)
Set the grob property specified by grob-property-path to value. grob-property-path is a symbol list of the form Context.GrobName.property or GrobName.property, possibly with subproperties given as well.

As opposed to \override which overrides the context-dependent defaults with which a grob is created, this command uses Output_property_engraver at the grob acknowledge stage. This may be necessary for overriding values set after the initial grob creation.
overrideTimeSignatureSettings [music] - time-signature (fraction, as pair) base-moment (fraction, as pair) beat-structure (list) beam-exceptions (list)
   Override timeSignatureSettings for time signatures of time-signature to have settings of base-moment, beat-structure, and beam-exceptions.

pageBreak [music]
   Force a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.

pageTurn [music]
   Force a page turn between two scores or top-level markups.

palmMute [music] - note (music)
   Print note with a triangle-shaped note head.

palmMuteOn [music]
   Set the default note head style to a triangle-shaped style.

parallelMusic [void] - voice-ids (list) music (music)
   Define parallel music sequences, separated by ’|’ (bar check signs), and assign them to the identifiers provided in voice-ids.
   voice-ids: a list of music identifiers (symbols containing only letters)
   music: a music sequence, containing BarChecks as limiting expressions.
   Example:
   \parallelMusic A,B,C {
     c c | d d | e e |
     d d | e e | f f |
   }
   A = { c c | d d }
   B = { d d | e e }
   C = { e e | f f }
   The last bar checks in a sequence are not copied to the result in order to facilitate ending the last entry at non-bar boundaries.

parenthesize [music] - arg (music)
   Tag arg to be parenthesized.

partcombine [music] - chord-range [pair of numbers] part1 (music) part2 (music)
   Take the music in part1 and part2 and return a music expression containing simultaneous voices, where part1 and part2 are combined into one voice where appropriate. Optional chord-range sets the distance in steps between notes that may be combined into a chord or unison.

partcombineDown [music] - chord-range [pair of numbers] part1 (music) part2 (music)
   Take the music in part1 and part2 and typeset so that they share a staff with stems directed downward.

partcombineForce [music] - type [symbol]
   Override the part-combiner.

partcombineUp [music] - chord-range [pair of numbers] part1 (music) part2 (music)
   Take the music in part1 and part2 and typeset so that they share a staff with stems directed upward.

partial [music] - dur (duration)
   Make a partial measure.
phrasingSlurDashPattern [music] - dash-fraction (number) dash-period (number)
Set up a custom style of dash pattern for dash-fraction ratio of line to space repeated at dash-period interval for phrasing slurs.

pitchedTrill [music] - main-note (music) secondary-note (music)
Print a trill with main-note as the main note of the trill and print secondary-note as a stemless note head in parentheses.

pointAndClickOff [void]
Suppress generating extra code in final-format (e.g. pdf) files to point back to the lilypond source statement.

pointAndClickOn [void]
Enable generation of code in final-format (e.g. pdf) files to reference the originating lilypond source statement; this is helpful when developing a score but generates bigger final-format files.

pointAndClickTypes [void] - types (symbol list or symbol)
Set a type or list of types (such as #'note-event) for which point-and-click info is generated.

propertyOverride [music] - grob-property-path (list of indexes or symbols) value (any type)
Set the grob property specified by grob-property-path to value. grob-property-path is a symbol list of the form Context.GrobName.property or GrobName.property, possibly with subproperties given as well. This music function is mostly intended for use from Scheme as a substitute for the built-in \override command.

propertyRevert [music] - grob-property-path (list of indexes or symbols)
Revert the grob property specified by grob-property-path to its previous value. grob-property-path is a symbol list of the form Context.GrobName.property or GrobName.property, possibly with subproperties given as well. This music function is mostly intended for use from Scheme as a substitute for the built-in \revert command.

propertySet [music] - property-path (symbol list or symbol) value (any type)
Set the context property specified by property-path to value. This music function is mostly intended for use from Scheme as a substitute for the built-in \set command.

propertyTweak [music] - prop (key list or symbol) value (any type) item (key list or music)
Add a tweak to the following item, usually music. This generally behaves like \tweak but will turn into an \override when item is a symbol list.
In that case, item specifies the grob path to override. This is mainly useful when using \propertyTweak as a component for building other functions like \omit. It is not the default behavior for \tweak since many input strings in \lyricmode can serve equally as music or as symbols which causes surprising behavior when tweaking lyrics using the less specific semantics of \propertyTweak.
prop can contain additional elements in which case a nested property (inside of an alist) is tweaked.

propertyUnset [music] - property-path (symbol list or symbol)
Unset the context property specified by property-path. This music function is mostly intended for use from Scheme as a substitute for the built-in \unset command.

pushToTag [music] - tag (symbol) more (music) music (music)
Add more to the front of elements of all music expressions in music that are tagged with tag.
quoteDuring [music] - what (string) main-music (music)
  Indicate a section of music to be quoted. what indicates the name of the quoted
  voice, as specified in an \addQuote command. main-music is used to indicate the
  length of music to be quoted; usually contains spacers or multi-measure rests.

reduceChords [music] - music (music)
  Reduce chords contained in music to single notes, intended mainly for reusing music
  in RhythmicStaff. Does not reduce parallel music.

relative [music] - pitch [pitch] music (music)
  Make music relative to pitch. If pitch is omitted, the first note in music is given in
  absolute pitch.

removeWithTag [music] - tags (symbol list or symbol) music (music)
  Remove elements of music that are tagged with one of the tags in tags. tags may
  be either a single symbol or a list of symbols.

resetRelativeOctave [music] - pitch (pitch)
  Set the octave inside a \relative section.

retrograde [music] - music (music)
  Return music in reverse order.

revertTimeSignatureSettings [music] - time-signature (pair)
  Revert timeSignatureSettings for time signatures of time-signature.

rightHandFinger [post event] - finger (number or markup)
  Apply finger as a fingering indication.

scaleDurations [music] - fraction (fraction, as pair) music (music)
  Multiply the duration of events in music by fraction.

settingsFrom (any type) - ctx [symbol] music (music)
  Take the layout instruction events from music, optionally restricted to those applying
  to context type ctx, and return a context modification duplicating their effect.

shape [music] - offsets (list) item (key list or music)
  Offset control-points of item by offsets. The argument is a list of number pairs or
  list of such lists. Each element of a pair represents an offset to one of the coordinates
  of a control-point. If item is a string, the result is \once\override for the specified
  grob type. If item is a music expression, the result is the same music expression with
  an appropriate tweak applied.

shiftDurations [music] - dur (integer) dots (integer) arg (music)
  Change the duration of arg by adding dur to the durlog of arg and dots to the
dots of arg.

single [music] - overrides (music) music (music)
  Convert overrides to tweaks and apply them to music. This does not convert
  \revert, \set or \unset.

skip [music] - dur (duration)
  Skip forward by dur.

slashedGrace [music] - music (music)
  Create slashed graces (slashes through stems, but no slur) from the following music
  expression

slurDashPattern [music] - dash-fraction (number) dash-period (number)
  Set up a custom style of dash pattern for dash-fraction ratio of line to space repeated
  at dash-period interval for slurs.
spacingTweaks [music] - parameters (list)
Set the system stretch, by reading the 'system-stretch property of the 'parameters' assoc list.

storePredefinedDiagram [void] - fretboard-table (hash table) chord (music) tuning (pair)
diagram-definition (string or pair)
Add predefined fret diagram defined by diagram-definition for the chord pitches chord and the string tuning tuning.

stringTuning (any type) - chord (music)
Convert chord to a string tuning. chord must be in absolute pitches and should have the highest string number (generally the lowest pitch) first.

styledNoteHeads [music] - style (symbol) heads (symbol list or symbol) music (music)
Set heads in music to style.

tabChordRepeats [music] - event-types [list] music (music)
Walk through music putting the notes, fingerings and string numbers of the previous chord into repeat chords, as well as an optional list of event-types such as #'(articulation-event).

tabChordRepetition [void]
Include the string and fingering information in a chord repetition. This function is deprecated; try using \tabChordRepeats instead.

tag [music] - tags (symbol list or symbol) music (music)
Tag the following music with tags and return the result, by adding the single symbol or symbol list tags to the tags property of music.

tagGroup [void] - tags (symbol list)
Define a tag group comprising the symbols in the symbol list tags. Tag groups must not overlap.

temporary [music] - music (music)
Make any \override in music replace an existing grob property value only temporarily, restoring the old value when a corresponding \revert is executed. This is achieved by clearing the 'pop-first' property normally set on \overrides.
An \override/\revert sequence created by using \temporary and \undo on the same music containing overrides will cancel out perfectly or cause a warning.
Non-property-related music is ignored, warnings are generated for any property-changing music that isn’t an \override.

tieDashPattern [music] - dash-fraction (number) dash-period (number)
Set up a custom style of dash pattern for dash-fraction ratio of line to space repeated at dash-period interval for ties.

time [music] - beat-structure [number list] fraction (fraction, as pair)
Set fraction as time signature, with optional number list beat-structure before it.

times [music] - fraction (fraction, as pair) music (music)
Scale music in time by fraction.

tocItem [music] - text (markup)
Add a line to the table of content, using the tocItemMarkup paper variable markup

transpose [music] - from (pitch) to (pitch) music (music)
Transpose music from pitch from to pitch to.
transposedCueDuring [music] - what (string) dir (direction) pitch (pitch) main-music (music)
Insert notes from the part what into a voice called cue, using the transposition defined by pitch. This happens simultaneously with main-music, which is usually a rest. The argument dir determines whether the cue notes should be notated as a first or second voice.

transposition [music] - pitch (pitch)
Set instrument transposition

tuplet [music] - ratio (fraction, as pair) tuplet-span [duration] music (music)
Scale the given music to tuplets. ratio is a fraction that specifies how many notes are played in place of the nominal value: it will be ‘3/2’ for triplets, namely three notes being played in place of two. If the optional duration tuplet-span is specified, it is used instead of tupletSpannerDuration for grouping the tuplets. For example,
\tuplet 3/2 4 { c8 c c c c c }
will result in two groups of three tuplets, each group lasting for a quarter note.

tupletSpan [music] - tuplet-span [duration]
Set tupletSpannerDuration, the length into which \tuplet without an explicit ‘tuplet-span’ argument of its own will group its tuplets, to the duration tuplet-span. To revert to the default of not subdividing the contents of a \tuplet command without explicit ‘tuplet-span’, use
\tupletSpan \default

tweak [music] - prop (key list or symbol) value (any type) music (music)
Add a tweak to the following music. Layout objects created by music get their property prop set to value. If prop has the form ‘Grob.property’, like with
\tweak Accidental.color #red cis'
an indirectly created grob (‘Accidental’ is caused by ‘NoteHead’) can be tweaked; otherwise only directly created grobs are affected.
prop can contain additional elements in which case a nested property (inside of an alist) is tweaked.

undo [music] - music (music)
Convert \override and \set in music to \revert and \unset, respectively. Any reverts and unsets already in music cause a warning. Non-property-related music is ignored.

unfoldRepeats [music] - types [symbol list or symbol] music (music)
Force \repeat volta, \repeat tremolo or \repeat percent commands in music to be interpreted as \repeat unfold, if specified in the optional symbol-list types. The default for types is an empty list, which will force any of those commands in music to be interpreted as \repeat unfold. Possible entries are volta, tremolo or percent. Multiple entries are possible.

voices [music] - ids (list of indexes or symbols) music (music)
Take the given key list of numbers (indicating the use of ‘\voiceOne’...) or symbols (indicating voice names, typically converted from strings by argument list processing) and assign the following \-separated music to contexts according to that list. Named rather than numbered contexts can be used for continuing one voice (for the sake of spanners and lyrics), usually requiring a \voiceOne-style override at the beginning of the passage and a \oneVoice override at its end.
The default
<< ... \ \ ... \ ... >>
construct would correspond to
\[ \text{voices 1,2,3} \ll ... \ \ll ... \ \ll ... \gg \]

void [void] - arg (any type)
Accept a scheme argument, return a void expression. Use this if you want to have a
scheme expression evaluated because of its side-effects, but its value ignored.

withMusicProperty [music] - sym (symbol) val (any type) music (music)
Set sym to val in music.

xNote [music] - note (music)
Print note with a cross-shaped note head.

\= [post event] - id (index or symbol) event (post event)
This sets the spanner-id property of the following event to the given id (non-
negative integer or symbol). This can be used to tell LilyPond how to connect
overlapping or parallel slurs or phrasing slurs within a single voice.

\fixed c' \{ c\=1{ d\=2( e\=1) f\=2) \}

\[ \]

A.19 Bezeichner zur Kontextveränderung

Folgende Befehle sind definiert, um Kontextveränderungen innerhalb von \layout oder \with
vorzunehmen:

RemoveAllEmptyStaves
Remove staves which are considered to be empty according to the list of interfaces
set by keepAliveInterfaces, including those in the first system.

- Sets grob property remove-empty in Abschnitt ‘‘VerticalAxisGroup’’ in
  Referenz der Interna to \#t.
- Sets grob property remove-first in Abschnitt ‘‘VerticalAxisGroup’’ in
  Referenz der Interna to \#t.

RemoveEmptyStaves
Remove staves which are considered to be empty according to the list of interfaces
set by keepAliveInterfaces.

- Sets grob property remove-empty in Abschnitt ‘‘VerticalAxisGroup’’ in
  Referenz der Interna to \#t.

A.20 Vordefinierte Typprädikate

R5RS primary predicates

<table>
<thead>
<tr>
<th>Type predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean?</td>
<td>boolean</td>
</tr>
<tr>
<td>char?</td>
<td>character</td>
</tr>
<tr>
<td>number?</td>
<td>number</td>
</tr>
<tr>
<td>pair?</td>
<td>pair</td>
</tr>
<tr>
<td>port?</td>
<td>port</td>
</tr>
<tr>
<td>procedure?</td>
<td>procedure</td>
</tr>
</tbody>
</table>
### Anhang A: Notationsübersicht

<table>
<thead>
<tr>
<th>string?</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbol?</td>
<td>symbol</td>
</tr>
<tr>
<td>vector?</td>
<td>vector</td>
</tr>
</tbody>
</table>

#### R5RS secondary predicates

<table>
<thead>
<tr>
<th>Type predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char-alphabetic?</td>
<td>alphabetic character</td>
</tr>
<tr>
<td>char-lower-case?</td>
<td>lower-case character</td>
</tr>
<tr>
<td>char-numeric?</td>
<td>numeric character</td>
</tr>
<tr>
<td>char-upper-case?</td>
<td>upper-case character</td>
</tr>
<tr>
<td>char-whitespace?</td>
<td>whitespace character</td>
</tr>
<tr>
<td>complex?</td>
<td>complex number</td>
</tr>
<tr>
<td>eof-object?</td>
<td>end-of-file object</td>
</tr>
<tr>
<td>even?</td>
<td>even number</td>
</tr>
<tr>
<td>exact?</td>
<td>exact number</td>
</tr>
<tr>
<td>inexact?</td>
<td>inexact number</td>
</tr>
<tr>
<td>input-port?</td>
<td>input port</td>
</tr>
<tr>
<td>integer?</td>
<td>integer</td>
</tr>
<tr>
<td>list?</td>
<td>list <em>(use cheap-list? for faster processing)</em></td>
</tr>
<tr>
<td>negative?</td>
<td>negative number</td>
</tr>
<tr>
<td>null?</td>
<td>null</td>
</tr>
<tr>
<td>odd?</td>
<td>odd number</td>
</tr>
<tr>
<td>output-port?</td>
<td>output port</td>
</tr>
<tr>
<td>positive?</td>
<td>positive number</td>
</tr>
<tr>
<td>rational?</td>
<td>rational number</td>
</tr>
<tr>
<td>real?</td>
<td>real number</td>
</tr>
<tr>
<td>zero?</td>
<td>zero</td>
</tr>
</tbody>
</table>

#### Guile predicates

<table>
<thead>
<tr>
<th>Type predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hash-table?</td>
<td>hash table</td>
</tr>
</tbody>
</table>

#### LilyPond scheme predicates

<table>
<thead>
<tr>
<th>Type predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean-or-symbol?</td>
<td>boolean or symbol</td>
</tr>
<tr>
<td>cheap-list?</td>
<td>list <em>(use this instead of list? for faster processing)</em></td>
</tr>
<tr>
<td>color?</td>
<td>color</td>
</tr>
<tr>
<td>fraction?</td>
<td>fraction, as pair</td>
</tr>
<tr>
<td>grob-list?</td>
<td>list of grobs</td>
</tr>
<tr>
<td>index?</td>
<td>non-negative integer</td>
</tr>
<tr>
<td>key?</td>
<td>index or symbol</td>
</tr>
<tr>
<td>key-list?</td>
<td>list of indexes or symbols</td>
</tr>
<tr>
<td>key-list-or-music?</td>
<td>key list or music</td>
</tr>
<tr>
<td>key-list-or-symbol?</td>
<td>key list or symbol</td>
</tr>
<tr>
<td>markup?</td>
<td>markup</td>
</tr>
<tr>
<td>markup-command-list?</td>
<td>markup command list</td>
</tr>
<tr>
<td>markup-list?</td>
<td>markup list</td>
</tr>
<tr>
<td>moment-pair?</td>
<td>pair of moment objects</td>
</tr>
</tbody>
</table>
LilyPond exported predicates

<table>
<thead>
<tr>
<th>Type predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ly:box?</td>
<td>box</td>
</tr>
<tr>
<td>ly:context?</td>
<td>context</td>
</tr>
<tr>
<td>ly:context-def?</td>
<td>context definition</td>
</tr>
<tr>
<td>ly:context-mod?</td>
<td>context modification</td>
</tr>
<tr>
<td>ly:dimension?</td>
<td>dimension, in staff space</td>
</tr>
<tr>
<td>ly:dir?</td>
<td>direction</td>
</tr>
<tr>
<td>ly:dispatcher?</td>
<td>dispatcher</td>
</tr>
<tr>
<td>ly:duration?</td>
<td>duration</td>
</tr>
<tr>
<td>ly:event?</td>
<td>post event</td>
</tr>
<tr>
<td>ly:font-metric?</td>
<td>font metric</td>
</tr>
<tr>
<td>ly:grob?</td>
<td>graphical (layout) object</td>
</tr>
<tr>
<td>ly:grob-array?</td>
<td>array of grobs</td>
</tr>
<tr>
<td>ly:grob-properties?</td>
<td>grob properties</td>
</tr>
<tr>
<td>ly:input-location?</td>
<td>input location</td>
</tr>
<tr>
<td>ly:item?</td>
<td>item</td>
</tr>
<tr>
<td>ly:iterator?</td>
<td>iterator</td>
</tr>
<tr>
<td>ly:lily-lexer?</td>
<td>lily-lexer</td>
</tr>
<tr>
<td>ly:lily-parser?</td>
<td>lily-parser</td>
</tr>
<tr>
<td>ly:listener?</td>
<td>listener</td>
</tr>
<tr>
<td>ly:moment?</td>
<td>moment</td>
</tr>
<tr>
<td>ly:music?</td>
<td>music</td>
</tr>
<tr>
<td>ly:music-function?</td>
<td>music function</td>
</tr>
<tr>
<td>ly:music-list?</td>
<td>list of music objects</td>
</tr>
<tr>
<td>ly:music-output?</td>
<td>music output</td>
</tr>
<tr>
<td>ly:otf-font?</td>
<td>OpenType font</td>
</tr>
<tr>
<td>ly:output-def?</td>
<td>output definition</td>
</tr>
<tr>
<td>ly:page-marker?</td>
<td>page marker</td>
</tr>
<tr>
<td>ly:pango-font?</td>
<td>pango font</td>
</tr>
</tbody>
</table>
ly:paper-system? paper-system Prob
ly:pitch? pitch
ly:prob? property object
ly:score? score
ly:skyline? skyline
ly:skyline-pair? pair of skylines
ly:source-file? source file
ly:spanner? spanner
ly:spring? spring
ly:stencil? stencil
ly:stream-event? stream event
ly:translator? translator
ly:translator-group? translator group
ly:undead? undead container
ly:unpure-pure-container? unpure/pure container

A.21 Scheme-Funktionen

ly:add-context-mod contextmods modification  [Funktion]
Add the given context modification to the list contextmods of context modifications.

ly:add-file-name-alist alist  [Funktion]
Add mappings for error messages from alist.

ly:add-interface iface desc props  [Funktion]
Add a new grob interface. iface is the interface name, desc is the interface description, and props is the list of user-settable properties for the interface.

ly:add-listener callback disp cl  [Funktion]
Add the single-argument procedure callback as listener to the dispatcher disp. Whenever disp hears an event of class cl, it calls callback with it.

ly:add-option sym val description  [Funktion]
Add a program option sym. val is the default value and description is a string description.

ly:all-grob-interfaces  [Funktion]
Return the hash table with all grob 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:angle x y  [Funktion]
Calculates angle in degrees of given vector. With one argument, x is a number pair indicating the vector. With two arguments, x and y specify the respective coordinates.

ly:assoc-get key alist default-value strict-checking  [Funktion]
Return value if key in alist, else default-value (or #f if not specified). If strict-checking is set to #t and key is not in alist, a programming error is output.

ly:axis-group-interface::add-element grob grob-element  [Funktion]
Set grob the parent of grob-element on all axes of grob.
ly:basic-progress str rest
A Scheme callable function to issue a basic progress message str. The message is formatted with format and rest.

ly:beam-score-count
count number of beam scores.

ly:book? x
Is x a Book object?


ly:book-add-score! book-smob score
Add score to book-smob score list.


Return header in book.

Return paper in book.

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).

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-scores book
Return scores in book.

Set the book header.

ly:box? x
Is x a Box object?

ly:bp num
num bigpoints (1/72th inch).

ly:bracket a iv t p
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
Send the stream event ev to the dispatcher disp.

ly:camel-case->lisp-identifier name-sym
Convert FooBar_Bla to foo-bar-bla style symbol.
**Funktion**

**ly:chain-assoc-get**<br>`key achain default-value strict-checking`<br>Return value for `key` from a list of alists `achain`. If no entry is found, return `default-value` or `#f` if `default-value` is not specified. With `strict-checking` set to `#t`, a programming error is output in such cases.

**Funktion**

**ly:check-expected-warnings**<br>Check whether all expected warnings have really been triggered.

**Funktion**

**ly:cm**<br>`num num cm`.<br>The Scheme code specified on command-line with `-e`.

**Funktion**

**ly:command-line-code**<br>The Scheme code specified on command-line with `-d`.

**Funktion**

**ly:command-line-options**<br>The Scheme options specified on command-line with `-d`.

**Funktion**

**ly:connect-dispatchers**<br>`to from`<br>Make the dispatcher `to` listen to events from `from`.

**Funktion**

**ly:context?**<br>`x`<br>Is `x` a Context object?

**Funktion**

**ly:context-current-moment**<br>`context`<br>Return the current moment of `context`.

**Funktion**

**ly:context-def?**<br>`x`<br>Is `x` a Context_def object?

**Funktion**

**ly:context-def-lookup**<br>`def sym val`<br>Return the value of `sym` in context definition `def` (e.g., \Voice). If no value is found, return `val` or `()` if `val` is undefined. `sym` can be any of `default-child`, `consists`, `description`, `aliases`, `accepts`, `property-ops`, `context-name`, `group-type`.

**Funktion**

**ly:context-def-modify**<br>`def mod`<br>Return the result of applying the context-mod `mod` to the context definition `def`. Does not change `def`.

**Funktion**

**ly:context-event-source**<br>`context`<br>Return `event-source` of context `context`.

**Funktion**

**ly:context-events-below**<br>`context`<br>Return a stream-distributor that distributes all events from `context` and all its subcontexts.

**Funktion**

**ly:context-find**<br>`context name`<br>Find a parent of `context` that has name or alias `name`. Return `#f` if not found.

**Funktion**

**ly:context-grob-definition**<br>`context name`<br>Return the definition of `name` (a symbol) within `context` as an alist.

**Funktion**

**ly:context-id**<br>`context`<br>Return the ID string of `context`, i.e., for `\context Voice = "one" ... return the string `one`.

**Funktion**

**ly:context-matched-pop-property**<br>`context grob cell`<br>This undoes a particular `\override \once \override` or `\once \revert` when given the specific alist pair to undo.
Anhang A: Notationsübersicht

ly:context-mod? x  
Is x a Context_mod object?  

ly:context-mod-apply! context mod  
Apply the context modification mod to context.

ly:context-name context  
Return the name of context, i.e., for \texttt{context Voice = "one" ...} return the symbol Voice.

ly:context-now context  
Return now-moment of context context.

ly:context-parent context  
Return the parent of context, \#f if none.

ly:context-property context sym def  
Return the value for property sym in context. If def is given, and property value is ', return def.

ly:context-property-where-defined context name  
Return the context above context where name is defined.

ly:context-pushpop-property context grob eltprop val  
Do \texttt{temporary \\overrid} or \texttt{\revert} operation in context. The grob definition grob is extended with eltprop (if val is specified) or reverted (if unspecified).

ly:context-set-property! context name val  
Set value of property name in context context to val.

ly:context-unset-property context name  
Unset value of property name in context context.

ly:debug str rest  
A Scheme callable function to issue a debug message str. The message is formatted with format and rest.

ly:default-scale  
Get the global default scale.

ly:dimension? d  
Return d as a number. Used to distinguish length variables from normal numbers.

ly:dir? s  
Is s a direction? Valid directions are -1, 0, or 1, where -1 represents left or down, 1 represents right or up, and 0 represents a neutral direction.

ly:directed direction magnitude  
Calculates an (x . y) pair with optional magnitude (defaulting to 1.0) and direction specified either as an angle in degrees or a coordinate pair giving the direction. If magnitude is a pair, the respective coordinates are scaled independently, useful for ellipse drawings.

ly:disconnect-dispatchers to from  
Stop the dispatcher to listening to events from from.

ly:dispatcher? x  
Is x a Dispatcher object?
ly:duration? x
  Is x a Duration object?

ly:duration<? p1 p2
  Is p1 shorter than p2?

ly:duration->string dur
  Convert dur to a string.

ly:duration-dot-count dur
  Extract the dot count from dur.

ly:duration-factor dur
  Extract the compression factor from dur. Return it as a pair.

ly:duration-length dur
  The length of the duration as a moment.

ly:duration-log dur
  Extract the duration log from dur.

ly:duration-scale dur
  Extract the compression factor from dur. Return it as a rational.

ly:effective-prefix
  Return effective prefix.

ly:encode-string-for-pdf str
  Encode the given string to either Latin1 (which is a subset of the PDFDocEncoding) or if
  that's not possible to full UTF-16BE with Byte-Order-Mark (BOM).

ly:engraver-announce-end-grob engraver grob cause
  Announce the end of a grob (i.e., the end of a spanner) originating from given engraver
  instance, with grob being a grob. cause should either be another grob or a music event.

ly:engraver-make-grob engraver grob-name cause
  Create a grob originating from given engraver instance, with given grob-name, a symbol.
  cause should either be another grob or a music event.

ly:error str rest
  A Scheme callable function to issue the error str. The error is formatted with format and
  rest.

ly:event? obj
  Is obj a proper (non-rhythmic) event object?

ly:event-deep-copy m
  Copy m and all sub expressions of m.

ly:event-property sev sym val
  Get the property sym of stream event sev. If sym is undefined, return val or '()' if val is not
  specified.

ly:event-set-property! ev sym val
  Set property sym in event ev to val.

ly:expand-environment str
  Expand $VAR and ${VAR} in str.
**ly:expect-warning str rest**

A Scheme callable function to register a warning to be expected and subsequently suppressed. If the warning is not encountered, a warning about the missing warning will be shown. The message should be translated with \(_ . . .\) and changing parameters given after the format string.

**ly:find-file name**

Return the absolute file name of name, or \#f if not found.

**ly:font-config-add-directory dir**

Add directory dir to FontConfig.

**ly:font-config-add-font font**

Add font font to FontConfig.

**ly:font-config-display-fonts**

Dump a list of all fonts visible to FontConfig.

**ly:font-config-get-font-file name**

Get the file for font name.

**ly:font-design-size font**

Given the font metric font, return the design size, relative to the current output-scale.

**ly:font-file-name font**

Given the font metric font, return the corresponding file name.

**ly:font-get-glyph font name**

Return a stencil from font for the glyph named name. If the glyph is not available, return an empty stencil.

Note that this command can only be used to access glyphs from fonts loaded with ly:system-font-load; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings fetaMusic and fetaBraces, respectively.

**ly:font-glyph-name-to-charcode font name**

Return the character code for glyph name in font.

Note that this command can only be used to access glyphs from fonts loaded with ly:system-font-load; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings fetaMusic and fetaBraces, respectively.

**ly:font-glyph-name-to-index font name**

Return the index for name in font.

Note that this command can only be used to access glyphs from fonts loaded with ly:system-font-load; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings fetaMusic and fetaBraces, respectively.

**ly:font-index-to-charcode font index**

Return the character code for index in font.

Note that this command can only be used to access glyphs from fonts loaded with ly:system-font-load; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings fetaMusic and fetaBraces, respectively.

**ly:font-magnification font**

Given the font metric font, return the magnification, relative to the current output-scale.
ly:font-metric? x  
Is x a Font_metric object?

ly:font-name font  
Given the font metric font, return the corresponding name.

ly:font-sub-fonts font  
Given the font metric font of an OpenType font, return the names of the subfonts within font.

ly:format str rest  
LilyPond specific format, supporting ~a and ~[0-9]f. Basic support for ~s is also provided.

ly:format-output context  
Given a global context in its final state, process it and return the Music_output object in its final state.

ly:generic-bound-extent grob common  
Determine the extent of grob relative to common along the X axis, finding its extent as a bound when it a has bound-alignment-interfaces property list set and otherwise the full extent.

ly:get-all-function-documentation  
Get a hash table with all LilyPond Scheme extension functions.

ly:get-all-translators  
Return a list of all translator objects that may be instantiated.

ly:get-cff-offset font-file-name idx  
Get the offset of 'CFF' table for font file name, returning it as an integer. The optional idx argument is useful for OpenType/CFF collections (OTC) only; it specifies the font index within the OTC. The default value of idx is 0.

ly:get-context-mods contextmod  
Returns the list of context modifications stored in contextmod.

ly:get-font-format font-file-name idx  
Get the font format for font_file_name, returning it as a symbol. The optional idx argument is useful for TrueType Collections (TTC) and OpenType/CFF collections (OTC) only; it specifies the font index within the TTC/OTC. The default value of idx is 0.

ly:get-option var  
Get a global option setting.

ly:get-spacing-spec from-scm to-scm  
Return the spacing spec going between the two given grobs, from_scm and to_scm.

ly:get-undead undead  
Get back object from undead.

ly:gettext original  
A Scheme wrapper function for gettext.

ly:grob? x  
Is x a Grob object?
**ly:grob-alist-chain** grob global

Get an alist chain for grob grob, with global as the global default. If unspecified, font-defaults from the layout block is taken.

**ly:grob-array?** x

Is x a Grob_array object?

**ly:grob-array->list** grob-arr

Return the elements of grob-arr as a Scheme list.

**ly:grob-array-length** grob-arr

Return the length of grob-arr.

**ly:grob-array-ref** grob-arr index

Retrieve the indexth element of grob-arr.

**ly:grob-basic-properties** grob

Get the immutable properties of grob.

**ly:grob-chain-callback** grob proc sym

Find the callback that is stored as property sym of grob grob and chain proc to the head of this, meaning that it is called using grob and the previous callback’s result.

**ly:grob-common-refpoint** grob other axis

Find the common refpoint of grob and other for axis.

**ly:grob-common-refpoint-of-array** grob others axis

Find the common refpoint of grob and others (a grob-array) for axis.

**ly:grob-default-font** grob

Return the default font for grob grob.

**ly:grob-extent** grob refp axis

Get the extent in axis direction of grob relative to the grob refp.

**ly:grob-get-vertical-axis-group-index** grob

Get the index of the vertical axis group the grob grob belongs to; return -1 if none is found.

**ly:grob-interfaces** grob

Return the interfaces list of grob grob.

**ly:grob-layout** grob

Get \layout definition from grob grob.

**ly:grob-object** grob sym

Return the value of a pointer in grob grob of property sym. It returns ’() (end-of-list) if sym is undefined in grob.

**ly:grob-original** grob

Return the unbroken original grob of grob.

**ly:grob-parent** grob axis

Get the parent of grob. axis is 0 for the X-axis, 1 for the Y-axis.

**ly:grob-pq<?** a b

Compare two grob priority queue entries. This is an internal function.
ly:grob-properties grob
Get the mutable properties of grob.

ly:grob-properties? x
Is x a Grob_properties object?

ly:grob-property grob sym val
Return the value for property sym of grob. If no value is found, return val or '()' if val is not specified.

ly:grob-property-data grob sym
Return the value for property sym of grob, but do not process callbacks.

ly:grob-pure-height grob refp beg end val
Return the pure height of grob given refpoint refp. If no value is found, return val or '()' if val is not specified.

ly:grob-pure-property grob sym beg end val
Return the pure value for property sym of grob. If no value is found, return val or '()' if val is not specified.

ly:grob-relative-coordinate grob refp axis
Get the coordinate in axis direction of grob relative to the grob refp.

ly:grob-robust-relative-extent grob refp axis
Get the extent in axis direction of grob relative to the grob refp, or (0,0) if empty.

ly:grob-script-priority-less a b
Compare two grobs by script priority. For internal use.

ly:grob-set-nested-property! grob symlist val
Set nested property symlist in grob grob to value val.

ly:grob-set-object! grob sym val
Set sym in grob grob to value val.

ly:grob-set-parent! grob axis parent-grob
Set parent-grob the parent of grob grob in axis axis.

ly:grob-set-property! grob sym val
Set sym in grob grob to value val.

ly:grob-spanned-rank-interval grob
Returns a pair with the rank of the furthest left column and the rank of the furthest right column spanned by grob.

ly:grob-staff-position sg
Return the Y-position of sg relative to the staff.

ly:grob-suicide! grob
Kill grob.

ly:grob-system grob
Return the system grob of grob.

ly:grob-translate-axis! grob d a
Translate grob on axis a over distance d.
ly:grob-vertical? a b  
Does a lie above b on the page?  

ly:gulp-file name size  
Read size characters from the file name, and return its contents in a string. If size is undefined, the entire file is read. The file is looked up using the search path.  

ly:has-glyph-names? font-file-name idx  
Does the font for font_file_name have glyph names? The optional idx argument is useful for TrueType Collections (TTC) and OpenType/CFF collections (OTC) only; it specifies the font index within the TTC/OTC. The default value of idx is 0.  

ly:hash-table-keys tab  
Return a list of keys in tab.  

ly:inch num  
num inches.  

ly:input-both-locations sip  
Return input location in sip as (file-name first-line first-column last-line last-column).  

ly:input-file-line-char-column sip  
Return input location in sip as (file-name line char column).  

ly:input-location? x  
Is x a Input object?  

ly:input-message sip msg rest  
Print msg as a GNU compliant error message, pointing to the location in sip. msg is interpreted similar to format’s argument, using rest.  

ly:input-warning sip msg rest  
Print msg as a GNU compliant warning message, pointing to the location in sip. msg is interpreted similar to format’s argument, using rest.  

ly:interpret-music-expression mus ctx  
Interpret the music expression mus in the global context ctx. The context is returned in its final state.  

ly:interpret-stencil-expression expr func arg1 offset  
Parse expr, feed bits to func with first arg arg1 having offset offset.  

ly:intlog2 d  
The 2-logarithm of 1/d.  

ly:item? g  
Is g an Item object?  

ly:item-break-dir it  
The break status direction of item it. -1 means end of line, 0 unbroken, and 1 beginning of line.  

ly:item-get-column it  
Return the PaperColumn or NonMusicalPaperColumn associated with this Item.  

ly:iterator? x  
Is x a Music_iterator object?
ly:length \ x \ y
[\text{Funktion}]
Calculates magnitude of given vector. With one argument, \( x \) is a number pair indicating the vector. With two arguments, \( x \) and \( y \) specify the respective coordinates.

ly:lexer-keywords \ lexer
[\text{Funktion}]
Return a list of (KEY . CODE) pairs, signifying the LilyPond reserved words list.

ly:lily-lexer? \ x
[\text{Funktion}]
Is \( x \) a Lily_lexer object?

ly:lily-parser? \ x
[\text{Funktion}]
Is \( x \) a Lily_parser object?

ly:line-interface::line \ grob \ startx \ starty \ endx \ endy
[\text{Funktion}]
Make a line using layout information from grob \( \text{grob} \).

ly:listened-event-class? \ disp \ cl
[\text{Funktion}]
Does \( \text{disp} \) listen to any event type in the list \( \text{cl} \)?

ly:listened-event-types \ disp
[\text{Funktion}]
Return a list of all event types that \( \text{disp} \) listens to.

ly:listener? \ x
[\text{Funktion}]
Is \( x \) a Listener object?

ly:make-book \ paper \ header \ scores
[\text{Funktion}]
Make a \textbackslash book of \( \text{paper} \) and \( \text{header} \) (which may be \#f as well) containing \textbackslash scores.

ly:make-book-part \ scores
[\text{Funktion}]
Make a \textbackslash bookpart containing \textbackslash scores.

ly:make-context-mod \ mod-list
[\text{Funktion}]
Creates a context modification, optionally initialized via the list of modifications \( \text{mod-list} \).

ly:make-dispatcher
[\text{Funktion}]
Return a newly created dispatcher.

ly:make-duration \ length \ dotcount \ num \ den
[\text{Funktion}]
\textit{length} 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 \textit{dotcount}.

The duration factor is optionally given by integers \textit{num} and \textit{den}, alternatively by a single rational number.

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.

ly:make-global-context \ output-def
[\text{Funktion}]
Set up a global interpretation context, using the output block \textit{output-def}. The context is returned.

ly:make-global-translator \ global
[\text{Funktion}]
Create a translator group and connect it to the global context \( \text{global} \). The translator group is returned.

ly:make-grob-properties \ alist
[\text{Funktion}]
This packages the given property list \( \text{alist} \) in a grob property container stored in a context property with the name of a grob.
ly:make-moment \( m \) \( g \) \( gn \) \( gd \)  
Create the moment with rational main timing \( m \), and optional grace timing \( g \).

A moment is a point in musical time. It consists of a pair of rationals \((m, g)\), where \( m \) is the timing for the main notes, and \( g \) the timing for grace notes. In absence of grace notes, \( g \) is zero.

For compatibility reasons, it is possible to write two numbers specifying numerator and denominator instead of the rationals. These forms cannot be mixed, and the two-argument form is disambiguated by the sign of the second argument: if it is positive, it can only be a denominator and not a grace timing.

ly:make-music \( props \)  
Make a C++ Music object and initialize it with \( props \).

This function is for internal use and is only called by \texttt{make-music}, which is the preferred interface for creating music objects.

ly:make-music-function \( signature \) \( func \)  
Make a function to process music, to be used for the parser. \texttt{func} is the function, and \texttt{signature} describes its arguments. \texttt{signature}'s cdr is a list containing either \texttt{ly:music?} predicates or other type predicates. Its car is the syntax function to call.

ly:make-music-relative! \( music \) \( pitch \)  
Make \( music \) relative to \( pitch \), return final pitch.

ly:make-output-def  
Make an output definition.

ly:make-page-label-marker \( label \)  
Return page marker with label \( label \).

ly:make-page-permission-marker \( symbol \) \( permission \)  
Return page marker with page breaking and turning permissions.

ly:make-pango-description-string \( chain \) \( size \)  
Make a \texttt{PangoFontDescription} string for the property alist \texttt{chain} at \texttt{size}.

ly:make-paper-outputter \( port \) \( format \)  
Create an outputter that evaluates within \texttt{output-format}, writing to \texttt{port}.

ly:make-pitch \( octave \) \( note \) \( alter \)  
\( octave \) is specified by an integer, zero for the octave containing middle C. \( note \) is a number indexing the global default scale, with 0 corresponding to pitch C and 6 usually corresponding to pitch B. Optional \( alter \) is a rational number of 200-cent whole tones for alteration.

ly:make-prob \( type \) \( init \) \( rest \)  
Create a \texttt{Prob} object.

ly:make-scale \( steps \)  
Create a scale. The argument is a vector of rational numbers, each of which represents the number of 200 cent tones of a pitch above the tonic.

ly:make-score \( music \)  
Return score with \texttt{music} encapsulated in it.

ly:make-spring \( ideal \) \( min-dist \)  
Make a spring. \( ideal \) is the ideal distance of the spring, and \( min-dist \) is the minimum distance.
**ly:make-stencil expr xext yext**  
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 `empty-interval` as its value), it is taken to be empty.

**ly:make-stream-event cl proplist**  
Create a stream event of class `cl` with the given mutable property list.

**ly:make-undead object**  
This packages `object` in a manner that keeps it from triggering "Parsed object should be dead" messages.

**ly:make-unpure-pure-container unpure pure**  
Make an unpure-pure container. `unpure` should be an unpure expression, and `pure` should be a pure expression. If `pure` is omitted, the value of `unpure` will be used twice, except that a callback is given two extra arguments that are ignored for the sake of pure calculations.

**ly:message str rest**  
A Scheme callable function to issue the message `str`. The message is formatted with `format` and `rest`.

**ly:minimal-breaking pb**  
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.

**ly:mm num**

```
num mm.
```

**ly:module->alist mod**  
Dump the contents of module `mod` as an alist.

**ly:module-copy dest src**  
Copy all bindings from module `src` into `dest`.

**ly:modules-lookup modules sym def**  
Look up `sym` in the list `modules`, returning the first occurrence. If not found, return `def` or `#f` if `def` isn’t specified.

**ly:moment? x**  
Is `x` a Moment object?

**ly:moment?>< a b**  
Compare two moments.

**ly:moment-add a b**  
Add two moments.

**ly:moment-div a b**  
Divide two moments.

**ly:moment-grace mom**  
Extract grace timing as a rational number from `mom`.

**ly:moment-grace-denominator mom**  
Extract denominator from grace timing.
**ly:moment-grace-numerator mom**
Extract numerator from grace timing.

**ly:moment-main mom**
Extract main timing as a rational number from mom.

**ly:moment-main-denominator mom**
Extract denominator from main timing.

**ly:moment-main-numerator mom**
Extract numerator from main timing.

**ly:moment-mod a b**
Modulo of two moments.

**ly:moment-mul a b**
Multiply two moments.

**ly:moment-sub a b**
Subtract two moments.

**ly:music? obj**
Is obj a music object?

**ly:music-compress m factor**
Compress music object m by moment factor.

**ly:music-deep-copy m origin**
Copy m and all sub expressions of m. m may be an arbitrary type; cons cells and music are copied recursively. If origin is given, it is used as the origin for one level of music by calling ly:set-origin! on the copy.

**ly:music-duration-compress mus fact**
Compress mus by factor fact, which is a Moment.

**ly:music-duration-length mus**
Extract the duration field from mus and return the length.

**ly:music-function? x**
Is x a Music_function object?

**ly:music-function-extract x**
Return the Scheme function inside x.

**ly:music-function-signature x**
Return the function signature inside x.

**ly:music-length mus**
Get the length of music expression mus and return it as a Moment object.

**ly:music-list? lst**
Is lst a list of music objects?

**ly:music-mutable-properties mus**
Return an alist containing the mutable properties of mus. The immutable properties are not available, since they are constant and initialized by the make-music function.
ly:music-output? x
   Is x a Music_output object? [Funktion]

ly:music-property mus sym val
   Return the value for property sym of music expression mus. If no value is found, return val
   or '() if val is not specified. [Funktion]

ly:music-set-property! mus sym val
   Set property sym in music expression mus to val. [Funktion]

ly:music-transpose m p
   Transpose m such that central C is mapped to p. Return m. [Funktion]

ly:note-column-accidentals note-column
   Return the AccidentalPlacement grob from note-column if any, or SCM_EOL otherwise. [Funktion]

ly:note-column-dot-column note-column
   Return the DotColumn grob from note-column if any, or SCM_EOL otherwise. [Funktion]

ly:note-head::stem-attachment font-metric glyph-name
   Get attachment in font-metric for attaching a stem to notehead glyph-name. [Funktion]

ly:number->string s
   Convert s to a string without generating many decimals. [Funktion]

ly:one-line-auto-height-breaking pb
   Put each score on a single line, and put each line on its own page. Modify the paper-width
   setting so that every page is wider than the widest line. Modify the paper-height setting to
   fit the height of the tallest line. [Funktion]

ly:one-line-breaking pb
   Put each score on a single line, and put each line on its own page. Modify the paper-width
   setting so that every page is wider than the widest line. [Funktion]

ly:one-page-breaking pb
   Put each score on a single page. The paper-height settings are modified so each score fits on
   one page, and the height of the page matches the height of the full score. [Funktion]

ly:optimal-breaking pb
   Optimally break (pages and lines) the Paper_book object pb to minimize badness in both
   vertical and horizontal spacing. [Funktion]

ly:option-usage port
   Print ly:set-option usage. Optional port argument for the destination defaults to current
   output port. [Funktion]

ly:otf->cff otf-file-name idx
   Convert the contents of an OTF file to a CFF file, returning it as a string. The optional
   idx argument is useful for OpenType/CFF collections (OTC) only; it specifies the font index
   within the OTC. The default value of idx is 0. [Funktion]

ly:otf-font? font
   Is font an OpenType font? [Funktion]

ly:otf-font-glyph-info font glyph
   Given the font metric font of an OpenType font, return the information about named glyph
   glyph (a string). [Funktion]
ly:otf-font-table-data font tag  
   Extract a table tag from font. Return empty string for non-existent tag.

ly:otf-glyph-count font  
   Return the number of glyphs in font.

ly:otf-glyph-list font  
   Return a list of glyph names for font.

ly:output-def? x  
   Is x a Output_def object?

ly:output-def-clone def  
   Clone output definition def.

ly:output-def-lookup def sym val  
   Return the value of sym in output definition def (e.g., \paper). If no value is found, return val or '()' if val is undefined.

ly:output-def-parent def  
   Return the parent output definition of def.

ly:output-def-scope def  
   Return the variable scope inside def.

ly:output-def-set-variable! def sym val  
   Set an output definition def variable sym to val.

ly:output-description output-def  
   Return the description of translators in output-def.

ly:output-find-context-def output-def context-name  
   Return an alist of all context defs (matching context-name if given) in output-def.

ly:output-formats  
   Formats passed to --format as a list of strings, used for the output.

ly:outputter-close outputter  
   Close port of outputter.

ly:outputter-dump-stencil outputter stencil  
   Dump stencil expr onto outputter.

ly:outputter-dump-string outputter str  
   Dump str onto outputter.

ly:outputter-module outputter  
   Return output module of outputter.

ly:outputter-output-scheme outputter expr  
   Eval expr in module of outputter.

ly:outputter-port outputter  
   Return output port for outputter.

ly:page-marker? x  
   Is x a Page_marker object?
ly:page-turn-breaking pb
Optimally break (pages and lines) the Paper_book object pb such that page turns only happen in specified places, returning its pages.

ly:pango-font? f
Is f a pango font?

ly:pango-font-physical-fonts f
Return alist of (ps-name file-name font-index) lists for Pango font f.

ly:paper-book? x
Is x a Paper_book object?

ly:paper-book-header pb
Return the header definition (\header) in Paper_book object pb.

ly:paper-book-pages pb

ly:paper-book-paper pb
Return the paper output definition (\paper) in Paper_book object pb.

ly:paper-book-performances pb

ly:paper-book-scopes pb

ly:paper-book-systems pb

ly:paper-column::break-align-width col align-syms
Determine the extent along the X-axis of a grob used for break-alignment organized by column col. The grob is specified by align-syms, which contains either a single break-align-symbol or a list of such symbols.

ly:paper-column::print
Optional stencil for PaperColumn or NonMusicalPaperColumn. Draws the rank number of each column, its moment in time, a blue arrow showing the ideal distance, and a red arrow showing the minimum distance between columns.

ly:paper-fonts def
Return a list containing the fonts from output definition def (e.g., \paper).

ly:paper-get-font def chain
Find a font metric in output definition def satisfying the font-qualifiers in alist chain chain, and return it. (An alist chain is a list of alists, containing grob properties.)

ly:paper-get-number def sym
Return the value of variable sym in output definition def as a double.

ly:paper-outputscale def
Return the output-scale for output definition def.

ly:paper-score-paper-systems paper-score
Return vector of paper_system objects from paper-score.
ly:paper-system? obj
Is obj a C++ Prob object of type paper-system?

ly:paper-system-minimum-distance sys1 sys2
Measure the minimum distance between these two paper-systems, using their stored skylines if possible and falling back to their extents otherwise.

ly:parse-file name

ly:parse-string-expression parser-smob ly-code filename line
Parse the string ly-code with parser-smob. Return the contained music expression. filename and line are optional source indicators.

ly:parsed-undead-list!
Return the list of objects that have been found live that should have been dead, and clear that list.

ly:parser-clear-error parser
Clear error flag for parser, defaulting to current parser.

ly:parser-clone closures location
Return a clone of current parser. An association list of port positions to closures can be specified in closures in order to have $ and # interpreted in their original lexical environment. If location is a valid location, it becomes the source of all music expressions inside.

ly:parser-define! symbol val
Bind symbol to val in current parser’s module.

ly:parser-error msg input
Display an error message and make current parser fail. Without a current parser, trigger an ordinary error.

ly:parser-has-error? parser
Does parser (defaulting to current parser) have an error flag?

ly:parser-include-string ly-code
Include the string ly-code into the input stream for current parser. Can only be used in immediate Scheme expressions ($ instead of #).

ly:parser-lexer parser
Return the lexer for parser, defaulting to current parser

ly:parser-lookup symbol
Look up symbol in current parser’s module. Return ’() if not defined.

ly:parser-output-name parser
Return the base name of the output file. If parser is left off, use currently active parser.

ly:parser-parse-string parser-smob ly-code
Parse the string ly-code with parser-smob. Upon failure, throw ly-file-failed key.

ly:parser-set-note-names names
Replace current note names in parser. names is an alist of symbols. This only has effect if the current mode is notes.

ly:performance-header performance
Return header of performance.
**ly:performance-set-header!**  *performance module*

Set the performance header.

**ly:performance-write**  *performance filename name*

Write *performance* to *filename* storing *name* as the name of the performance in the file metadata.

**ly:pitch?**  *x*

Is *x* a *Pitch* object?

**ly:pitch<?**  *p1 p2*

Is *p1* lexicographically smaller than *p2*?

**ly:pitch-alteration**  *pp*

Extract the alteration from pitch *pp*.

**ly:pitch-diff**  *pitch root*

Return pitch *delta* such that *root* transposed by *delta* equals *pitch*.

**ly:pitch-negate**  *p*

Negate *p*.

**ly:pitch-notename**  *pp*

Extract the note name from pitch *pp*.

**ly:pitch-octave**  *pp*

Extract the octave from pitch *pp*.

**ly:pitch-quartertones**  *pp*

Calculate the number of quarter tones of *pp* from middle C.

**ly:pitch-semitones**  *pp*

Calculate the number of semitones of *pp* from middle C.

**ly:pitch-steps**  *p*

Number of steps counted from middle C of the pitch *p*.

**ly:pitch-tones**  *pp*

Calculate the number of tones of *pp* from middle C as a rational number.

**ly:pitch-transpose**  *p delta*

Transpose *p* by the amount *delta*, where *delta* is relative to middle C.

**ly:pointer-group-interface::add-grob**  *grob sym grob-element*

Add *grob-element* to *grob*’s *sym* grob array.

**ly:position-on-line?**  *sg spos*

Return whether *spos* is on a line of the staff associated with the grob *sg* (even on an extender line).

**ly:prob?**  *x*

Is *x* a *Prob* object?

**ly:prob-immutable-properties**  *prob*

Retrieve an alist of immutable properties.

**ly:prob-mutable-properties**  *prob*

Retrieve an alist of mutable properties.
**ly:prob-property** prob sym val

Return the value for property sym of Prob object prob. If no value is found, return val or '()' if val is not specified.

**ly:prob-property?** obj sym

Is boolean prop sym of sym set?

**ly:prob-set-property!** obj sym value

Set property sym of obj to value.

**ly:prob-type?** obj type

Is obj the specified prob-type?

**ly:programming-error** str rest

A Scheme callable function to issue the internal warning str. The message is formatted with format and rest.

**ly:progress** str rest

A Scheme callable function to print progress str. The message is formatted with format and rest.

**ly:property-lookup-stats** sym

Return hash table with a property access corresponding to sym. Choices are prob, grob, and context.

**ly:protects**

Return hash of protected objects.

**ly:pt** num

num printer points.

**ly:round-filled-box** xext yext blot

Make a Stencil object that prints a black box of dimensions xext, yext and roundness blot.

**ly:round-filled-polygon** points blot extroversion

Make a Stencil object that prints a black polygon with corners at the points defined by points (list of coordinate pairs) and roundness blot. Optional extroversion shifts the outline outward, with the default of -1.0 keeping the outer boundary of the outline just inside of the polygon.
ly:run-translator mus output-def
[Funktion]
Process mus according to output-def. An interpretation context is set up, and mus 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.

ly:score? x
[Funktion]
Is x a Score object?

ly:score-add-output-def! score def
[Funktion]
Add an output definition def to score.

ly:score-embedded-format score layout
[Funktion]
Run score through layout (an output definition) scaled to correct output-scale already, returning a list of layout-lines.

ly:score-error? score
[Funktion]
Was there an error in the score?

ly:score-header score
[Funktion]
Return score header.

ly:score-music score
[Funktion]
Return score music.

ly:score-embedded-format score layout
[Funktion]
Run score through layout (an output definition) scaled to correct output-scale already, returning a list of layout-lines.

ly:score-error? score
[Funktion]
Was there an error in the score?

ly:score-error? score
[Funktion]
Was there an error in the score?

ly:score-error? score
[Funktion]
Was there an error in the score?

ly:score-error? score
[Funktion]
Was there an error in the score?

ly:score-error? score
[Funktion]
Was there an error in the score?
**ly:set-origin!** $m$ origin  
[**Funktion**]  
This sets the origin given in origin to $m$. $m$ will typically be a music expression or a list of music. List structures are searched recursively, but recursion stops at the changed music expressions themselves. origin is generally of type ly:input-location?, defaulting to (*location*). Other valid values for origin are a music expression which is then used as the source of location information, or #f or '()' in which case no action is performed. The return value is $m$ itself.

**ly:set-property-cache-callback** $cb$  
[**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.

**ly:skyline?** $x$  
[**Funktion**]  
Is $x$ a Skyline object?

**ly:skyline-empty?** $sky$  
[**Funktion**]  
Return whether $sky$ is empty.

**ly:skyline-pair?** $x$  
[**Funktion**]  
Is $x$ a Skyline_pair object?

**ly:slur-score-count**  
[**Funktion**]  
Count number of slur scores.

**ly:smob-protects**  
[**Funktion**]  
Return LilyPond’s internal smob protection list.

**ly:solve-spring-rod-problem** $springs$ $rods$ $length$ ragged  
[**Funktion**]  
Solve a spring and rod problem for count objects, that are connected by count-1 springs, and an arbitrary number of rods. count is implicitly given by springs and rods. The springs argument has the format (ideal, inverse_hook) and rods is of the form (idx1, idx2, distance).

length is a number, ragged a boolean.

The function returns a list containing the force (positive for stretching, negative for compressing and #f for non-satisfied constraints) followed by spring-count+1 positions of the objects.

**ly:source-file?** $x$  
[**Funktion**]  
Is $x$ a Source_file object?

**ly:source-files** parser-smob  
[**Funktion**]  
A list of LilyPond files being processed; a PARSER may optionally be specified.

**ly:spanner?** $g$  
[**Funktion**]  
Is $g$ a spanner object?

**ly:spanner-bound** $spanner$ dir  
[**Funktion**]  
Get one of the bounds of spanner. dir is -1 for left, and 1 for right.

**ly:spanner-broken-into** $spanner$  
[**Funktion**]  
Return broken-into list for spanner.

**ly:spanner-set-bound!** $spanner$ dir item  
[**Funktion**]  
Set grob item as bound in direction dir for spanner.
ly:spawn command rest
   [Funktion]
   Simple interface to g.spawn sync str. The error is formatted with format and rest.

ly:spring? x
   Is x a Spring object?

ly:spring-set-inverse-compress-strength! spring strength
   [Funktion]
   Set the inverse compress strength of spring.

ly:spring-set-inverse-stretch-strength! spring strength
   [Funktion]
   Set the inverse stretch strength of spring.

ly:staff-symbol-line-thickness grob
   [Funktion]
   Returns the current staff-line thickness in the staff associated with grob, expressed as a multiple of the current staff-space height.

ly:staff-symbol-staff-radius grob
   [Funktion]
   Returns the radius of the staff associated with grob.

ly:staff-symbol-staff-space grob
   [Funktion]
   Returns the current staff-space height in the staff associated with grob, expressed as a multiple of the default height of a staff-space in the traditional five-line staff.

ly:start-environment
   [Funktion]
   Return the environment (a list of strings) that was in effect at program start.

ly:stderr-redirect file-name mode
   [Funktion]
   Redirect stderr to file-name, opened with mode.

ly:stencil? x
   Is x a Stencil object?

ly:stencil-add args
   [Funktion]
   Combine stencils. Takes any number of arguments.

ly:stencil-aligned-to stil axis dir
   [Funktion]
   Align stil using its own extents. dir is a number. -1 and 1 are left and right, respectively. Other values are interpolated (so 0 means the center).

ly:stencil-combine-at-edge first axis direction second padding
   [Funktion]
   Construct a stencil by putting second next to first. axis can be 0 (x-axis) or 1 (y-axis). direction can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with padding as extra space. first and second may also be ’() or #f.

ly:stencil-empty? stil axis
   [Funktion]
   Return whether stil is empty. If an optional axis is supplied, the emptiness check is restricted to that axis.

ly:stencil-expr stil
   [Funktion]
   Return the expression of stil.

ly:stencil-extent stil axis
   [Funktion]
   Return a pair of numbers signifying the extent of stil in axis direction (0 or 1 for x and y axis, respectively).

ly:stencil-fonts s
   [Funktion]
   Analyze s, and return a list of fonts used in s.
ly:stencil-in-color stc r g b
   [Funktion]
   Put stc in a different color.

ly:stencil-outline stil outline
   [Funktion]
   Return a stencil with the stencil expression (inking) of stencil stil but with outline and dimensions from stencil outline.

ly:stencil-rotate stil angle x y
   [Funktion]
   Return a stencil stil rotated angle degrees around the relative offset (x, y). E.g., an offset of (-1, 1) will rotate the stencil around the left upper corner.

ly:stencil-rotate-absolute stil angle x y
   [Funktion]
   Return a stencil stil rotated angle degrees around point (x, y), given in absolute coordinates.

ly:stencil-scale stil x y
   [Funktion]
   Scale stencil stil using the horizontal and vertical scaling factors x and y. Negative values will flip or mirror stil without changing its origin; this may result in collisions unless it is repositioned.

ly:stencil-stack first axis direction second padding mindist
   [Funktion]
   Construct a stencil by stacking second next to first. axis can be 0 (x-axis) or 1 (y-axis). direction can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with padding as extra space. first and second may also be '()' or '#f'. As opposed to ly:stencil-combine-at-edge, metrics are suited for successively accumulating lines of stencils. Also, second stencil is drawn last.
   If mindist is specified, reference points are placed apart at least by this distance. If either of the stencils is spacing, padding and mindist do not apply.

ly:stencil-translate stil offset
   [Funktion]
   Return a stil, but translated by offset (a pair of numbers).

ly:stencil-translate-axis stil amount axis
   [Funktion]
   Return a copy of stencil but translated by amount in axis direction.

ly:stream-event? obj
   [Funktion]
   Is obj a Stream_event object?

ly:string-percent-encode str
   [Funktion]
   Encode all characters in string str with hexadecimal percent escape sequences, with the following exceptions: characters -, ., /, and _; and characters in ranges 0-9, A-Z, and a-z.

ly:string-substitute a b s
   [Funktion]
   Replace string a by string b in string s.

ly:system-font-load name
   [Funktion]
   Load the OpenType system font name.otf. Fonts loaded with this command must contain three additional SFNT font tables called LILC, LILF, and LILY, needed for typesetting musical elements. Currently, only the Emmentaler and the Emmentaler-Brace fonts fulfill these requirements.
   Note that only ly:font-get-glyph and derived code (like \lookup) can access glyphs from the system fonts; text strings are handled exclusively via the Pango interface.

ly:text-interface::interpret-markup
   [Funktion]
   Convert a text markup into a stencil. Takes three arguments, layout, props, and markup. layout is a \layout block; it may be obtained from a grob with ly:grob-layout. props is an alist chain, i.e. a list of alists. This is typically obtained with (ly:grob-alist-chain grob (ly:output-def-lookup layout 'text-font-defaults)). markup is the markup text to be processed.
**ly:translate-cpp-warning-scheme str**  
Funktion  
Translates a string in C++ printf format and modifies it to use it for scheme formatting.

**ly:translator? x**  
Funktion  
Is x a Translator object?

**ly:translator-context trans**  
Funktion  
Return the context of the translator object trans.

**ly:translator-description creator**  
Funktion  
Return an alist of properties of translator definition creator.

**ly:translator-group? x**  
Funktion  
Is x a Translator_group object?

**ly:translator-name creator**  
Funktion  
Return the type name of the translator definition creator. The name is a symbol.

**ly:transpose-key-alist l pit**  
Funktion  
Make a new key alist of l transposed by pitch pit.

**ly:truncate-list! lst i**  
Funktion  
Take at most the first i of list lst.

**ly:ttf->pfa ttf-file-name idx**  
Funktion  
Convert the contents of a TrueType font file to PostScript Type 42 font, returning it as a string. The optional idx argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of idx is 0.

**ly:ttf-ps-name ttf-file-name idx**  
Funktion  
Extract the PostScript name from a TrueType font. The optional idx argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of idx is 0.

**ly:type1->pfa type1-file-name**  
Funktion  
Convert the contents of a Type 1 font in PFB format to PFA format. If the file is already in PFA format, pass through it.

**ly:undead? x**  
Funktion  
Is x a Undead object?

**ly:unit**  
Funktion  
Return the unit used for lengths as a string.

**ly:unpure-call data grob rest**  
Funktion  
Convert property data (unpure-pure container or procedure) to value in an unpure context defined by grob and possibly rest arguments.

**ly:unpure-pure-container? x**  
Funktion  
Is x a Unpure_pure_container object?

**ly:unpure-pure-container-pure-part pc**  
Funktion  
Return the pure part of pc.

**ly:unpure-pure-container-unpure-part pc**  
Funktion  
Return the unpure part of pc.
ly:usage
   Print usage message.

ly:verbose-output?
   Was verbose output requested, i.e. loglevel at least DEBUG?

ly:version
   Return the current lilypond version as a list, e.g., (1 3 127 uu1).

ly:warning str rest
   A Scheme callable function to issue the warning str. The message is formatted with format and rest.

ly:warning-located location str rest
   A Scheme callable function to issue the warning str at the specified location in an input file. The message is formatted with format and rest.

ly:wide-char->utf-8 wc
   Encode the Unicode codepoint wc, an integer, as UTF-8.
Anhang B Befehlsübersicht

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Erklärung</th>
<th>Beispiel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 8 16</td>
<td>Tondauern</td>
<td><img src="image1.png" alt="Tondauern" /></td>
</tr>
<tr>
<td>c4. c4..</td>
<td>Punktierung</td>
<td><img src="image2.png" alt="Punktierung" /></td>
</tr>
<tr>
<td>c d e f g a b</td>
<td>Tonleiter</td>
<td><img src="image3.png" alt="Tonleiter" /></td>
</tr>
<tr>
<td>fis bes</td>
<td>Vorzeichen</td>
<td><img src="image4.png" alt="Vorzeichen" /></td>
</tr>
<tr>
<td>\clef treble \clef bass</td>
<td>Notenschlüssel</td>
<td><img src="image5.png" alt="Notenschlüssel" /></td>
</tr>
<tr>
<td>\time 3/4 \time 4/4</td>
<td>Taktangaben</td>
<td><img src="image6.png" alt="Taktangaben" /></td>
</tr>
<tr>
<td>r4 r8</td>
<td>Pause</td>
<td><img src="image7.png" alt="Pause" /></td>
</tr>
<tr>
<td>d − d</td>
<td>Bindebogen</td>
<td><img src="image8.png" alt="Bindebogen" /></td>
</tr>
<tr>
<td>\key es \major</td>
<td>Tonart</td>
<td><img src="image9.png" alt="Tonart" /></td>
</tr>
</tbody>
</table>
note' Oktavierung

note, Oktavierung nach unten

c\(d\ e\) Legatobogen

c\(c\(d\) e\) Phrasierungsboßen

a8[ b] Balken

<< \new Staff . . . >> mehr Notensysteme

c\(-> c\.-\) Artikulationszeichen

c2\mf c\sfz Dynamik

a\(< a\) a\! Crescendo
a\> a a\!

Decrescendo

< >

Noten im Akkord

\partial 8

Auftakt

\tuplet 3/2 \{f g a\}

Triolen

\grace

Verzierungen

\lyricmode \{ twinkle \}

Texteingabe
twinkle

\new Lyrics

Gesangstext
twinkle

twin -- kle

Gesangstext-Trennstrich
twin - kle

\chordmode \{ c:dim f:maj7 \}

Akkorde

\context ChordNames

Akkordsymbole drucken

C° F△

<<\{e f\} \ \{c d\}>>

Mehrstimmigkeit
unsichtbare Pausen
Anhang C GNU Free Documentation License

Version 1.3, 3 November 2008

http://fsf.org/

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 functional and useful 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. Secondarily, 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, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

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. (Thus, 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. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

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 Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.
A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document 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, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. 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, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include 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, PostScript or PDF 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.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

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 (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must encode 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 computer-network location from which the general
network-using public has access to download using public-standard network protocols a
complete Transparent copy of the Document, free of added material. 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 fewer than
five), unless they release you from this requirement.

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 copy-
right 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”, Preserve 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. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements 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 to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

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, and that you preserve all their Warranty Disclaimers.

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 “Acknowledgements”, 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, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which 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 half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket 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, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation. Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.
Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.
ADDENDUM: How to use this License for your documents

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 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License''.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being list their titles, with the Front-Cover Texts being list, and with the Back-Cover Texts being list.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

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 D 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.

! ........................................ 6, 116  ? ........................................ 6
" ........................................ 102  [ ........................................ 88
"|" ........................................ 102  [ ........................................ 88
, ........................................ 2  ] ........................................ 88
\ ........................................ 126  ] ........................................ 88
) ........................................ 126  ^ ....................................... 392
\ ........................................ 126  _ ....................................... 249
. ........................................ 42  _ ....................................... 113
/ ........................................ 392  _ ....................................... 102
/\ ........................................ 392  ^ ....................................... 392
: ........................................ 150  ~ ....................................... 49
< ........................................ 116, 152  \< .................................... 116, 152
<..< .................................... 152  \< .................................... 116, 152
= ........................................ 9, 756  >= .................................... 116, 152
> ........................................ 
\> .................................... 116, 152
Anhang D: Index der LilyPond-Befehle

A
\abs-fontsize ................................................. 225, 646
absolute ....................................................... 745
\accepts ....................................................... 113
\accepts ....................................................... 565, 566, 567
\accidentals ................................................. 105, 745
\accStyle ....................................................... 24, 745
\addChordShape .............................................. 350, 745
\addInstrumentDefinition .................................. 194, 202, 745
additionalPitchPrefix ....................................... 397
\addlyrics ..................................................... 244, 245, 246
\addQuote ...................................................... 195, 745
\aeolian ......................................................... 20
\afterGrace .................................................... 106, 745
\aikenHeads ................................................... 37
\aikenHeadsMinor ............................................ 38
\alias .......................................................... 565
alignAboveContext ........................................... 568
alignBelowContext ........................................... 265, 568
\allowPageTurn ............................................... 516, 745
allowVoltaHook ............................................... 745
alterBroken ..................................................... 745
\annotate-spacing ........................................... 549
\appendToTag ............................................... 479, 746
applyContext .................................................. 746
applyMusic ..................................................... 746
applyOutput .................................................... 746
\appoggiatura ............................................... 106, 746
\arrow ........................................................ 130
\arrowDown ..................................................... 130
\arrowUp ......................................................... 130
\arrowBracket ................................................. 131
\arrowParentheses ............................................. 131
\arrowParentesisDashed ..................................... 131
\arrowhead ..................................................... 233, 672
\ascendens .................................................... 426, 432
assertBeamQuant ............................................... 746
assertBeamSlope ............................................... 746
\auctum ......................................................... 426, 432
aug ..................................................... 389
\augment ....................................................... 432
\auto-first-page-number ..................................... 507
\auto-footnote ............................................... 694
autoBeaming ................................................... 80
\autoBeamOff .................................................. 77, 308
\autoBeamOn .................................................... 77
\autoBeamSettings ............................................. 85
\autochange .................................................... 306, 746
\automaticBars ................................................. 594
\backslashed-digit ........................................... 694
Balloon_engraver ............................................. 211
\balloonGroBText ............................................. 211, 746
balloonLengthOff ............................................ 211
balloonLengthOn ............................................. 211
\balloonText ................................................... 211, 746
banjo-c-tuning ................................................. 365
banjo-modal-tuning ............................................ 365
banjo-open-d-tuning .......................................... 365
banjo-open-dm-tuning ......................................... 365
\bar .......................................................... 92, 97, 746
barCheckSynchronize ......................................... 102
\barNumber ...................................................... 97
\barNumberCheck ............................................... 102, 746
\barNumberVisibility ......................................... 97
\barType ........................................................ 97
\base ........................................................... 537
\baseMoment .................................................... 80
\beam .......................................................... 672
beamExceptions ............................................... 80, 746
\beatStructure ................................................ 80
\bendAfter ..................................................... 128, 746
\binding-off ................................................... 505
\blank ........................................................ 506
\blank-after-score-page-penalty ............................ 506
\blank-last-page-penalty ..................................... 506
\blank-page-penalty ........................................... 506
\bold .......................................................... 225, 647
\book ......................................................... 448, 451
\bookOutputName ............................................. 450, 746
\bookOutputSuffix ............................................ 450, 746
\bookpart ...................................................... 449, 514, 514
\bottomMargin ................................................ 501
\box .......................................................... 231, 647
\bracket ......................................................... 121, 231, 310, 672
\break ......................................................... 514
\break-align-symbols ......................................... 599
\break-visibility ............................................... 590
breakable ....................................................... 78
\breakbefore ................................................... 458
\breathe ....................................................... 127, 746
\breve ........................................................ 41, 52
\cadenzaoff .................................................... 69
\cadenzaOn ..................................................... 69
\caesura ....................................................... 424
\caps .......................................................... 647
\cavum ........................................................ 426, 432
\center-align .................................................. 228, 656
\center-column ................................................ 229, 657
\change ......................................................... 305
\char ........................................................ 695
check-consistency ............................................ 504
chordChanges .................................................. 395
\chordMode .................................................... 5, 13, 348
\chordNameExceptions ....................................... 398
\chordNameLowerCaseMinor .................................. 396
ChordNames ..................................................... 348, 393
\chordNameSeparator ......................................... 397
\chordNoteName ................................................ 397
\chordPrefixSpacer ............................................ 398
\chordRepeats ................................................ 324, 747
\chordRootName ............................................... 396
\chords ......................................................... 394
\circle ......................................................... 231, 673
\clef .......................................................... 16, 747
\cm ........................................................... 583
\coda .......................................................... 113
\color ........................................................ 208
\column ........................................................ 229, 657
\columnlines ................................................... 701
\combine ....................................................... 233, 657
### Anhang D: Index der LilyPond-Befehle

<table>
<thead>
<tr>
<th>Befehl</th>
<th>Zeile</th>
</tr>
</thead>
<tbody>
<tr>
<td>common-shortest-duration</td>
<td>537</td>
</tr>
<tr>
<td>completion_heads_engraver</td>
<td>74</td>
</tr>
<tr>
<td>completion_rest_engraver</td>
<td>74</td>
</tr>
<tr>
<td>\compound-meter</td>
<td>680</td>
</tr>
<tr>
<td>\compoundMeter</td>
<td>73, 747</td>
</tr>
<tr>
<td>\compressFullBarRests</td>
<td>56, 58</td>
</tr>
<tr>
<td>\compressMMRests</td>
<td>747</td>
</tr>
<tr>
<td>\concat</td>
<td>658</td>
</tr>
<tr>
<td>\consists</td>
<td>565</td>
</tr>
<tr>
<td>\context</td>
<td>554, 560</td>
</tr>
<tr>
<td>\controlpitch</td>
<td>9</td>
</tr>
<tr>
<td>\cresc</td>
<td>116</td>
</tr>
<tr>
<td>\crescHairpin</td>
<td>117</td>
</tr>
<tr>
<td>\crescTextCresc</td>
<td>117</td>
</tr>
<tr>
<td>\cross</td>
<td>34</td>
</tr>
<tr>
<td>\cross-staff</td>
<td>308</td>
</tr>
<tr>
<td>\crossStaff</td>
<td>747</td>
</tr>
<tr>
<td>\cueClef</td>
<td>198, 747</td>
</tr>
<tr>
<td>\cueClefUnset</td>
<td>747</td>
</tr>
<tr>
<td>\cueDuring</td>
<td>198, 747</td>
</tr>
<tr>
<td>\cueDuringWithClef</td>
<td>198, 747</td>
</tr>
<tr>
<td>\CueVoice</td>
<td>198</td>
</tr>
<tr>
<td>\currentBarNumber</td>
<td>97, 111</td>
</tr>
<tr>
<td>\customTabClef</td>
<td>681</td>
</tr>
<tr>
<td>\deacc</td>
<td>24</td>
</tr>
<tr>
<td>\defaultBarType</td>
<td>97</td>
</tr>
<tr>
<td>\defaultTimeSignature</td>
<td>61</td>
</tr>
<tr>
<td>\defineBarLine</td>
<td>747</td>
</tr>
<tr>
<td>\deminutum</td>
<td>426, 432</td>
</tr>
<tr>
<td>\denotes</td>
<td>565, 566, 567</td>
</tr>
<tr>
<td>\descend</td>
<td>426, 432</td>
</tr>
<tr>
<td>\dim</td>
<td>117, 389</td>
</tr>
<tr>
<td>\dimHairpin</td>
<td>117</td>
</tr>
<tr>
<td>\dimTextDecr</td>
<td>117</td>
</tr>
<tr>
<td>\dimTextDecresc</td>
<td>117</td>
</tr>
<tr>
<td>\dimTextDim</td>
<td>117</td>
</tr>
<tr>
<td>\dir-column</td>
<td>658</td>
</tr>
<tr>
<td>\discont</td>
<td>689</td>
</tr>
<tr>
<td>\displayLilyMusic</td>
<td>496, 747</td>
</tr>
<tr>
<td>\displayMusic</td>
<td>747</td>
</tr>
<tr>
<td>\displayScheme</td>
<td>747</td>
</tr>
<tr>
<td>\divisions</td>
<td>424</td>
</tr>
<tr>
<td>\divisionsMaxima</td>
<td>424</td>
</tr>
<tr>
<td>\divisionsMinima</td>
<td>424</td>
</tr>
<tr>
<td>\dodecaphonic</td>
<td>29</td>
</tr>
<tr>
<td>\dorion</td>
<td>20</td>
</tr>
<tr>
<td>\dots</td>
<td>42</td>
</tr>
<tr>
<td>\dotsNeutral</td>
<td>42</td>
</tr>
<tr>
<td>\dotsUp</td>
<td>42</td>
</tr>
<tr>
<td>\doubleflat</td>
<td>681</td>
</tr>
<tr>
<td>\doublesharpe</td>
<td>681</td>
</tr>
<tr>
<td>\downbow</td>
<td>113, 317</td>
</tr>
<tr>
<td>\downmordent</td>
<td>113</td>
</tr>
<tr>
<td>\downrall</td>
<td>113</td>
</tr>
<tr>
<td>\draw-circle</td>
<td>233, 673</td>
</tr>
<tr>
<td>\draw-dashed-line</td>
<td>673</td>
</tr>
<tr>
<td>\draw-dotted-line</td>
<td>674</td>
</tr>
</tbody>
</table>

### E

<table>
<thead>
<tr>
<th>Befehl</th>
<th>Zeile</th>
</tr>
</thead>
<tbody>
<tr>
<td>\easyHeadsOff</td>
<td>35</td>
</tr>
<tr>
<td>\easyHeadsOn</td>
<td>35</td>
</tr>
<tr>
<td>\ellipse</td>
<td>675</td>
</tr>
<tr>
<td>\endspangers</td>
<td>747</td>
</tr>
<tr>
<td>\epsfile</td>
<td>233, 676</td>
</tr>
<tr>
<td>\expresivo</td>
<td>113, 117</td>
</tr>
<tr>
<td>\eventChords</td>
<td>747</td>
</tr>
<tr>
<td>\expandFullBarRests</td>
<td>56, 58</td>
</tr>
<tr>
<td>\explicitClefVisibility</td>
<td>592</td>
</tr>
<tr>
<td>\explicitKeySignatureVisibility</td>
<td>592</td>
</tr>
<tr>
<td>\extra-offset</td>
<td>521</td>
</tr>
<tr>
<td>\eyeglasses</td>
<td>695</td>
</tr>
</tbody>
</table>

### F

<table>
<thead>
<tr>
<th>Befehl</th>
<th>Zeile</th>
</tr>
</thead>
<tbody>
<tr>
<td>\fill</td>
<td>116</td>
</tr>
<tr>
<td>\fff</td>
<td>116</td>
</tr>
<tr>
<td>\fff</td>
<td>116</td>
</tr>
<tr>
<td>\fill-line</td>
<td>230, 659</td>
</tr>
<tr>
<td>\fill-with-pattern</td>
<td>659</td>
</tr>
<tr>
<td>\filled-box</td>
<td>233, 676</td>
</tr>
<tr>
<td>\finalis</td>
<td>424</td>
</tr>
<tr>
<td>\finger</td>
<td>205, 648, 748</td>
</tr>
<tr>
<td>\first-page-number</td>
<td>597</td>
</tr>
<tr>
<td>\first-visible</td>
<td>607</td>
</tr>
<tr>
<td>\fixed</td>
<td>748</td>
</tr>
<tr>
<td>\flageollet</td>
<td>113</td>
</tr>
<tr>
<td>\flat</td>
<td>681</td>
</tr>
<tr>
<td>\flexa</td>
<td>432</td>
</tr>
<tr>
<td>\followVoice</td>
<td>308</td>
</tr>
<tr>
<td>\font-interface</td>
<td>204, 237</td>
</tr>
<tr>
<td>\font-size</td>
<td>204</td>
</tr>
<tr>
<td>-fontCaps</td>
<td>648</td>
</tr>
<tr>
<td>\fontsize</td>
<td>225, 648</td>
</tr>
<tr>
<td>\fontSize</td>
<td>204</td>
</tr>
<tr>
<td>\footnote</td>
<td>695, 748</td>
</tr>
<tr>
<td>\forget</td>
<td>30</td>
</tr>
<tr>
<td>\four-string-banjo</td>
<td>365</td>
</tr>
<tr>
<td>\fp</td>
<td>116</td>
</tr>
<tr>
<td>\fraction</td>
<td>695</td>
</tr>
<tr>
<td>\freeBass</td>
<td>690</td>
</tr>
<tr>
<td>\frenchChords</td>
<td>398</td>
</tr>
<tr>
<td>\fret-diagram</td>
<td>339, 686</td>
</tr>
<tr>
<td>\fret-diagram-interface</td>
<td>343</td>
</tr>
<tr>
<td>\fret-diagram-terse</td>
<td>341, 686</td>
</tr>
<tr>
<td>\fret-diagram-verbose</td>
<td>342, 687</td>
</tr>
<tr>
<td>\FretBoards</td>
<td>346</td>
</tr>
</tbody>
</table>
Anhang D: Index der LilyPond-Befehle

\fromproperty ................................................. 696
\funkHeads .................................................. 37
\funkHeadsMinor ................................................ 38

G
\general-align ................................................. 229, 660
\germanChords .................................................. 398
\glissando ...................................................... 129
\grace ......................................................... 105, 748
Grid_line_span_engraver ......................................... 174
gridInterval .................................................... 212
grobdirection .................................................. 748
grobdirection .................................................. 91

H
\halfopen ....................................................... 113
\halign ......................................................... 228, 661
\harmonic ....................................................... 318, 326
\harmonicByFret ................................................. 326, 748
\harmonicByRatio ............................................... 326, 748
\harmonicNote ................................................... 748
\harmonicsOn .................................................... 748
\harp-pedal ...................................................... 688
\bracket .......................................................... 231, 676
\center-in ....................................................... 662
\header .......................................................... 451
\hide ........................................................... 748
\hideKeySignature ................................................. 377
\hideNotes ........................................................ 207
\hideSplitTiedTabNotes .......................................... 325
\hideStaffSwitch ................................................. 308
\horizontal-shift ............................................... 505
\horizontal السلطات ............................................ 214
\hspace .......................................................... 662
\huge ........................................................... 204, 227, 648

I
\improvisationOff ............................................. 40, 75
\improvisationOn ................................................. 40, 75
\in ............................................................... 583
\incipit ........................................................ 748
\inclinatum ..................................................... 426, 432
\include ........................................................ 476
\include-settings ................................................. 482
\indent .......................................................... 193, 505, 541
\inner-acceptability ............................................ 749
\inner-margin .................................................... 505
\inStaffSegno ...................................................... 139, 749
\instrumentSwitch ................................................ 194, 749
\inversion ........................................................ 13, 749
\ionian ........................................................... 20
\italianChords ................................................... 398
\italic .......................................................... 225, 649

J
\justified-lines ................................................. 230, 701
\justify .......................................................... 230, 664
\justify-field ................................................... 663
\justify-line .................................................... 663
\justify-string ................................................... 664

K
\keepWithTag .................................................... 479, 749
\key .............................................................. 20, 38, 749
KievanStaff ....................................................... 433
KievanVoice ....................................................... 433
\killCues ........................................................ 202, 749

L
\label ............................................................ 749
\label ............................................................ 474
\laissezVibrer .................................................. 50
landscape ......................................................... 499
language .......................................................... 749
languageRestore .................................................. 749
languageSaveAndChange .......................................... 749
\large ............................................................ 204, 227, 649
\larger ........................................................... 225, 227, 649
\last-bottom-spacing ............................................ 503
\layout .......................................................... 451, 509, 560
\layoutfile ....................................................... 511
\left-align ....................................................... 228, 665
\left-brace ....................................................... 696
\left-column ...................................................... 665
left-margin ....................................................... 504
\heel ............................................................. 113
\line ............................................................. 665
\line-width ....................................................... 504, 541
\line ............................................................. 426, 432
\lineprall ......................................................... 113
\locrian ......................................................... 20
\longa ........................................................... 41, 52
\longfletch ..................................................... 113
\lookup .......................................................... 696
\lower ........................................................... 228, 666
\ltoe .............................................................. 113
\ly:minimal-breaking ............................................ 516
\ly:one-line-breaking ........................................... 517
\ly:optimal-breaking ........................................... 515
\ly:page-turn-breaking .......................................... 515
\ly:diatonic ..................................................... 20
\lyricmode ....................................................... 242, 244
\lyricsto ......................................................... 244, 245, 246
Anhang D: Index der LilyPond-Befehle

M

\magnify ................................................. 389
\magnifyMusic .......................... 749
\magnifyStaff .......................... 749
\maj ................................................. 20, 204, 583
\major ................................................. 20
\major seven symbols .................. 398
\majorSevenSymbol ....................... 396
\makeDynamicScript ..................... 121
\makePangoFontTree ...................... 239
\makeClusters ......................... 157, 749
\makeDefaultStringTuning ............... 749
\markCommand .................................. 701
\marcato ........................................... 113
\mark ........................................ 102, 219, 749
\markAlphabet .......................... 697
\markLetter ...................................... 697
\markup ........................................... 219, 222, 223
\markUpMarkUpSpacing ................. 503
\markUpSystemSpacing ................. 503
\markupMap ...................................... 749
\maxSystemPerPage ....................... 506
\maxima ........................................... 41, 52
\measureLength .......................... 80, 111
\measurePosition .................... 68, 111
\medium .......................................... 650
\melisma ......................................... 250
\melismaEnd ................................... 250
\mensuralStaff ......................... 174, 414
\mensuralVoice ......................... 414
\mergeDifferentlyDottedOff ........... 161
\mergeDifferentlyDottedOn ............ 161
\mergeDifferentlyHeadedOff ........... 161
\mergeDifferentlyHeadedOn ............ 161
\mf ............................................... 116
\midi ............................................. 451
\minSystemsPerPage ..................... 506
\minimumYExtent ......................... 521
\minimumFret .................................. 323, 358
\minimumPageTurnLength ............... 516
\minimumRepeatLengthForPageTurn ...... 516
\minor ............................................. 20
\minorChordModifier .................... 398
\mixed .......................................... 310
\mixolydian .................................. 20
\mm ............................................... 583
\modalInversion ......................... 15, 750
\modalTranspose ....................... 14, 750
\modern ......................................... 26
\modernCautionary .................... 27
\modernVoice ................................ 27
\modernVoiceCautionary ................ 27
\moderntab .................................. 337
\moderntext .................................. 113
\mp ............................................... 116
\multiMeasureRestText ................ 57
\musicGlyph .................................. 104, 681
\musicMap ...................................... 750

N
\name .............................................. 565
\natural .......................................... 682
\neoModern ..................................... 28
\neoModernCautionary ................... 28
\neoModernVoice .......................... 29
\neoModernVoiceCautionary ............ 29
\new ............................................. 554
\noReset ........................................ 30
\noBeam .......................................... 89
\noBreak ........................................ 514
\nonStaffNonStaffSpacing .............. 521
\nonStaffRelatedStaffSpacing .......... 521
\nonStaffUnrelatedStaffSpacing ....... 521
\noPageBreak ................................. 515, 750
\noPageTurn .................................. 516, 750
\normalSizeSub .............................. 650
\normalSizeSuper .......................... 220, 650
\normalText .................................... 650
\normalSize .................................... 204, 227, 651
\note ............................................. 682
\noteByNumber ................................ 682
\noteHeadsEngraver ....................... 74
\null ............................................. 228, 697
\numerator ..................................... 651
\numericTimeSignature ................... 61

O
\octaveCheck ..................................... 9, 750
\offset ............................................. 750
\omit .............................................. 750
\onTheFly ....................................... 464, 697
\once ............................................. 574, 575, 750
\oneVoice ....................................... 157
\open ............................................. 113, 317
\oriscus ........................................ 420, 432
\ottava ......................................... 22, 750
\outerMargin .................................. 505
\outsideStaffHorizontalPadding ........ 535
\outsideStaffPadding .................... 535
\outsideStaffPriority .................... 535
\oval ............................................. 677
\overlay .......................................... 666
\override ........................................ 574, 697
\override in \lyricmode .................. 242
\overrideLines ............................... 702
\overrideProperty ......................... 750
\overrideTimeSignatureSettings ...... 61, 751
\overtie ......................................... 651

P
\p .................................................... 116
\padAround ...................................... 232, 666
\padMarkup ...................................... 232, 666
\padToBox ....................................... 232, 667
\padX ............................................. 232, 667
\pageBreaking .................................. 507
\pageBreakingSystemSystemSpacing ...... 507
\pageCount ...................................... 507
\pagelink ...................................... 698
\pageRef ......................................... 698
\pageRef ......................................... 474
Anhang D: Index der LilyPond-Befehle

propertySet ........................................... 508
propertyTweak ........................................ 752
propertyUnset ........................................ 752
\pt ...................................................... 598
\pushToTag ............................................ 479, 752
\put-adjacent ......................................... 667

\quilisma ............................................... 426, 432
quotedCueEventTypes .................................. 197
quotedEventTypes ...................................... 197
\quoteDuring .......................................... 195, 198, 753

\r ........................................................ 52
ragged-bottom ......................................... 501
ragged-last ............................................. 505, 541
ragged-last-bottom ................................... 501
ragged-right ............................................ 504, 541
\raise ................................................... 228, 667
reduceChords ........................................... 753
\relative ............................................... 2, 5, 13, 307, 753
\RemoveEmptyStaves .................................. 188, 189
\removeWithTag ....................................... 479, 753
\repeat .................................................. 136
\repeat percent ....................................... 147
\repeat tremolo ....................................... 149
\repeatCommands ....................................... 143
\repeatTie .............................................. 49, 140, 267
\replace ............................................... 652
\resetRelativeOctave ................................ 753
\rest ..................................................... 52, 683
\ rests-by-number .................................... 683
\revert .................................................. 14, 753
\reverseTurn .......................................... 113
\revert .................................................. 575
\revertTimeSignatureSettings ....................... 63, 753
\rfz ...................................................... 116
\rgb-color ............................................ 200
\rheel ................................................... 113
RhythmicStaff ......................................... 174
\right-align ........................................... 228, 668
\right-brace .......................................... 698
\right-column ......................................... 698
right-margin .......................................... 501
\rightHandFinger ...................................... 359, 753
\roman ................................................... 652
\rotate .................................................. 668
\rounded-box ......................................... 231, 679
\rtoe ................................................... 113
R .......................................................... 56
S

\sacrificedHarps
\sacrificedHarpsMinor
\sacc.
\scale
\scaleDurations
\core
\core-lines
\score-markup-spacing
\score-system-spacing
\scoreTitleMarkup
\segno
\self-alignment-X
\semiflat
\semigrants
\sestitus
\set-octavation
\settingsFrom
\af
\aff
\afz
\shape
\sharp
\shiftDurations
\shifff
\shiftUn
\shiftOn
\shiftOnn
\short-indent
\shortfermata
\show-available-fonts
\showFirstLength
\showKeySignature
\showLastLength
\showStaffSwitch
\signaturecongruentiae
\simple
\single
\skip
\slashTypeSetting
\slashChordSeparator
\slashdigit
\slashGrace
\slashDashed
\slashDashPattern
\slashDotted
\slashDown
\slashHalfDashed
\slashHalfSolid
\slashNeutral
\slashSolid
\slashUp
\small
\smallCaps
\smaller
\sapplicati
to
\stemUp
\stemGroup
\stemStaff
\startGroup
\startTrillSpan
\stop
\stopStaff
\stopTrillSpan
\store
\storePredefinedDiagram
\stringTuning
\stringTunings
\stropha
\strut
\styledNoteheads
\sub
\subdivideBeams
\suggestAccidentals
\super
\sustain
\sustainOn
\system-count
\system-separator-markup
\system-system-spacing
\tabChordRepeats
\tabChordRepetition
\tabFullNotation
\tabl
\table-of-contents
\tabStaff
\TabVoice
\tag
\tagGroup
\taylor
\teaching
\teeny
Anhang D: Index der LilyPond-Befehle

\tempo ........................................ 65
temporary ..................................... 754
\tenuto ........................................ 113
\text .......................................... 310, 654
\textLengthOff ................................ 58, 217
\textLengthOn ................................ 58, 217
textSpannerDown ................................ 218
textSpannerNeutral ................................ 218
textSpannerUp ................................... 218
\thumb ........................................... 113, 205
\tie ............................................. 654
\tied-Lyric ..................................... 685
\tieDashed ....................................... 50
tieDashPattern .................................. 754
\tieDotted ....................................... 50
\tieDown ......................................... 50
\tieNeutral ....................................... 50
\tieSolid ......................................... 50
\tieUp ........................................... 50
\time .............................................. 60, 80, 754
\times ............................................ 43, 71, 754
timeSignatureFraction ............................. 71
\tiny .............................................. 204, 227, 655
\tocItem .......................................... 476, 754
top-margin ....................................... 501
top-markup-spacing ................................. 503
top-system-spacing ................................. 503
\translate ......................................... 229, 669
\translate-scaled ................................ 229, 669
\transparent ..................................... 699
\transpose ......................................... 5, 10, 13, 754
\transposedCueDuring ............................. 204, 755
\transposition ................................... 23, 195, 755
\treCorde .......................................... 310
tremolo ............................................ 149
\textLength ......................................... 150
\triangle ........................................... 233, 680
\trill ............................................. 113, 133
tuplet .............................................. 755
\tupletDown ....................................... 44
\tupletNeutral ................................... 44
TupletNumber ....................................... 45
tupletNumberFormatFunction ....................... 44
tupletSpan ......................................... 755
tupletSpannerDuration ............................. 44
tupletUp ........................................... 44
\turn .............................................. 113
\weak ............................................... 576, 755
two-sided ......................................... 505
\type ............................................... 565
\typewriter ....................................... 655
\universal .......................................... 655
\unaCorda ......................................... 310
\underline .......................................... 225, 655
\undertie .......................................... 656
\undo ................................................ 755
\unfold ............................................. 145
\unfoldRepeats .................................... 491, 755
\unhideNotes ...................................... 207
\unset .............................................. 573
\upbow ............................................. 113, 317
\upmordent ....................................... 113
\upprall .......................................... 113
\upright .......................................... 656
\varcodas .......................................... 113
VaticanStaff ....................................... 174, 421
VaticanVoice ...................................... 174, 421
\vcenter .......................................... 669
\verbatim-file .................................... 699
VerticalAxisGroup ................................ 521
\vertLongfermata ................................ 113
\virga ............................................ 420, 432
\virgula ........................................... 424
\voice ............................................. 24, 26
\voiceFourStyle ................................... 161
\voiceNeutralStyle ................................ 161
\voiceOne ......................................... 157
\voiceOne .. \voiceFour ................................ 157
\voiceOneStyle .................................... 161
\voices ........................................... 755
\voiceThreeStyle .................................. 161
\voiceTwoStyle .................................... 161
\void ............................................... 496, 756
\vspace ............................................ 669
\walkerHeads ...................................... 37
\walkerHeadsMinor ................................ 38
\whichBar ......................................... 97
\whiteout ......................................... 700
\with ............................................... 558, 563
\with-color ........................................ 208, 700
\with-dimensions .................................. 701
\with-dimensions-from ............................. 701
\with-link ......................................... 701
\with-line ......................................... 701
\withOutline ....................................... 701
\with-url ........................................... 680
\withMusicProperty ................................ 756
\woodwind-diagram ................................ 689
\wordwrap .......................................... 230, 671
\wordwrap-field ................................... 670
\wordwrap-internal ................................ 703
\wordwrap-lines ................................... 230, 703
\wordwrap-string .................................. 671
\wordwrap-string-internal ......................... 703
\x11-color ........................................ 208, 209
X-offset .......................................... 521
xNote ............................................. 756
Anhang E  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.

| !                     | 6, 116 |
| !                      | 6      |
| "                     | 102    |
| "|"                    | 88     |
| 2                      | 88     |
| (                      | 392    |
| \                      | 126    |
| )                      | 249    |
| \                      | 102    |
| 2                      | 49     |
| !                      | 42     |
| /                      | 392    |
| /+                     | 392    |
| :                      | 150    |
| \<                    | 116, 152 |
| <...>                  | 152    |
| =                      | 49     |
| *                      | 756    |
| >                      | 22     |
| \>                     | 116, 152 |

Ändern von Instrumentenbezeichnungen ........... 193
Ändern von Schriftarten für das gesamte Dokument ........... 239
Äolisch ........... 20
Über die Nicht-Schachtelung von Klamern und Bindebögen ........... 125, 127
Überbindung ........... 49
Überbindung in Wiederholung ........... 140
Überbindung und Wiederholungen ........... 49
Überbindung, Versetzungssymbole ........... 6
Überbindungen und Akkorde ........... 49
Überlappende Notation in Ordnung bringen ........... 305, 306
Überspringen von Noten im Gesangstext ........... 266
Überspringen von Zeichen ........... 54
Übungszeichen ........... 102
Übungszeichen formatieren ........... 103
Übungszeichenstil ........... 103
Übungsziele, Notenköpfe ........... 35

15ma ........... 22
Anhang E: LilyPond-Index

8

8va ......................... 22
8ve ......................... 22

A

a due .................. 170
a due-Stellen ........ 166
Abbildungen im Text 231
\abs-fontsize .......... 225, 646
Abschnitte definieren, Notenabstände 538
Abschnitte markieren 102
absolute ................ 745
absolute Lautstärke 116
Absolute Spezifikation von Oktaven 1
Absoluter Modus: Tonhöhen 1
Abstände, absolut 583
Abstände, skaliert 583
Abstände, vertikal 520
Abstand vergrößern, Gesangstext 260
Abstand von Hilfslinien 181
Abstand zwischen Notensystemen 292
Abstand zwischen Systemen in Klaviernoten 308
Abstrich .................. 113, 705
Accelerando in MIDI 490
\accent .................. 113, 115
Accentus ................ 705
\accepts .............. 565, 566, 567
\accliacatura .......... 105, 109, 745
accidental ............. 419, 423, 434
Accidental ............ 7, 30
accidental-interface ... 7
accidental-suggestion-interface 30
Accidental_engraver 7, 6, 240
AccidentalCautionary 7
AccidentalPlacement 30
\accidentalStyle ....... 24, 745
AccidentalSuggestion 30, 420
\addChordShape....... 350, 745
adding a white background to text 700
\addInstrumentDefinition 194, 202, 745
additionalPitchPrefix 307
Additionen in Akkorden 391
\addlyrics ............. 244, 245, 246
\addquote .............. 195, 745
\aeolian .............. 20
Aeolisch .............. 20
\afterGrace .......... 106, 745
Aken-Notenkörpfe 37
\aikenHeads .......... 37
\aikenHeadsMinor ... 38
Akolade ................ 175
Akkord, eine Noten verändern 576
Akkord, gebrochen 130
Akkord-Diagramm .... 346
Akkordbezeichnungen 388, 393
Akkordbezeichnungen und Bunddiagramme 348
Akkorddiagramm ....... 338
Akkorddiagramme, automatisch 337
Akkorde .................. 152, 388
Akkorde über zwei Systeme 308
Akkorde und Überblendungen 49
Akkorde und relativer Modus 4
Akkorde, Entfernen von Tönen 392
Akkorde, relative Tonhöhe 153
Akkorde, Unterdrückung wiederholt 395
Akkorde, Wiederholung 154, 324
Akkorde, zwischen Systemen mit \autochange 308
Akkorde farbige Noten 209
Akkorde Fingersatz 205
Akkorde Versetzungszeichen 30
Akkordegenschaften 389
Akkordeon ............. 311
Akkordeon, Diskant-Symbole 311
Akkordeon, Register 311
Akkordformen für buyierte Saiteninstrumente 350
Akkordformen für Bondinstrumente 350
Akkordgliissando ....... 326
Akkordmodus ......... 389
Akkordmodus ........... 388
Akkordstufen, Alteration 392
Akkordstufen, Veränderung 392
Akkordsymbole ..... 393
Akkordsymbole in MIDI 490
Akkordsymbole, anpassen 396
Akkordtabulatur ....... 338
Akzent .................. 113, 705
Akzidentien ........... 5, 24
al niente ............. 119, 121
alias ................. 565
alignAboveContext .... 568
alignBelowContext .... 265, 568
alista ................. 708
\All layout objects .... 572, 590, 709
\allowPageTurn ....... 516, 745
allowVoltaHook ....... 745
Alte Schlüssel .......... 16
alterBroken ........... 745
\alternative .......... 136
alternative Endungen und Gesangstext 266
alternative Schlüsse in ausgeschriebenen Wiederholungen 145
Alternative Schlüsse mit Bindebogen 140
alternativer Gesangstext 270
alternativer Schluss 136
Altschlüssel ........... 16
Ambitus ............ 31, 33
ambitus ............ 33, 242
ambitus-interface ... 34
Ambitus_engraver .... 33
AmbitusAccidental ........ 33
AmbitusLine ......... 33
AmbitusNoteHead ...... 34
anacrusis ............. 69
Analyse ............ 214
\ancient notation ....... 413, 418, 425
Andere Benutzung von Optimierungen 304
dergleichen Stimmen zitieren 198
Anfänger, Notenlernen 35
Anführungstrich im Text 224
Anführungstrich, Gesangstext 249
Anführungszeichen, Gesangstext 242
Angabe der Oktave: absolut 1
Anmerkung, Blase 211
annotate-spacing ...... 549
Anordnung, horizontal .... 536
Anpassen von Akkordsymbolen 306
Anpassen von Bunddiagrammen 343
Anpassen von staff symbol 583
Anpassungen im Gesangstextmodus \[ 242 \]
Anstrich \[ 113 \]
Anzahl der Notenlinien einstellen \[ 181 \]
Anzahl der Wiederholung, ändern \[ 143 \]
\appendToTag \[ \text{479, 746} \]
applyContext \[ 746 \]
applyMusic \[ 746 \]
applyOutput \[ 746 \]
\appogiatura \[ 105, 109, 746 \]
arabische Musik \[ 410 \]
arabische Musik, Beispiel \[ 444 \]
arabische Notenbezeichnungen \[ 441 \]
arabische Taktarten \[ 443 \]
arabische Tonarten \[ 442 \]
arabische Vorziehen \[ 442 \]
arabisches Halb-B Versetzungszeichen \[ 441 \]
Arbeiten an Eingabe-Dateien \[ 448 \]
Arpeggio \[ 130, 133 \]
\arpeggio \[ 130, 133 \]
Arpeggio über Systeme im Klammernstil \[ 133 \]
Arpeggio-Symbole, besondere \[ 131 \]
\arpeggio\ArrowDown \[ 130 \]
\arpeggio\ArrowUp \[ 130 \]
\arpeggio\Bracket \[ 131 \]
\arpeggio\Normal \[ 130 \]
\arpeggio\Parenthesis \[ 131 \]
\arpeggio\ParenthesisDashed \[ 131 \]
\arrow-head \[ 233, 672 \]
Art der Übungszeichen \[ 103 \]
Arten von Notenköpfen \[ 646 \]
articulation-event \[ 197 \]
Artikulationszeichen \[ 113 \]
Artikulationszeichen und Lautstärke \[ 121 \]
Artikulationszeichen, greg. Choral \[ 424 \]
Artikulier-Skript \[ 490 \]
\ascends .. \[ \text{426, 432} \]
assertBeamQuant \[ 746 \]
assertBeamSlope \[ 746 \]
associateVoice \[ 241 \]
assoziative Liste \[ 708 \]
Atemzeichen \[ 127 \]
\auctum \[ 426, 432 \]
Aufführungsanweisung: Tempo \[ 65 \]
Ausklinkungen von wiederholten Noten \[ 145 \]
Ausschlusszeichen \[ 5 \]
Auftisch \[ 705 \]
Auftakt \[ 68 \]
Auftakt in Wiederholung \[ 138 \]
Aufteilen von Noten \[ 74 \]
Aufteilen von Pausen \[ 74 \]
aug \[ 389 \]
\augmentum \[ 432 \]
Auszehlen von Noten \[ 48 \]
Auszüge, Text \[ 223 \]
Ausgabe von Akkordbezeichnungen \[ 393 \]
ausgeschriebene Wiederholungen \[ 145 \]
Ausklingen lassen \[ 50 \]
Ausklingen lassen, Bögen \[ 50 \]
Auslassene von Noten im Gesangstext \[ 266 \]
Ausnahmen, Akkordsymbole \[ 398 \]
Ausrichten an Kadenz \[ 110 \]
Ausrichten an Objekten \[ 599 \]
Ausrufung von Gesangstext \[ 245 \]
Ausrufung von Taktlinien \[ 100 \]
Ausrufung von Text \[ 228 \]
Ausrichtung von Text, Befehle \[ 231 \]
Ausrichtung, Papier \[ 499 \]
Aussehen von Taktzahlen \[ 98 \]
Auswahl von Schriftgröße (Notation) \[ 204 \]
auto-first-page-number \[ \text{507} \]
auto-footnote \[ 694 \]
auto-knee-gap \[ 78 \]
Auto_beam_engraver \[ 79, 87 \]
autobeam \[ 80 \]
\autoBearing \[ \text{77, 308} \]
\autoBeamOff \[ 77 \]
\autoBeamMx \[ 77 \]
\autoBeamSettings \[ 85 \]
\autochange \[ \text{300, 746} \]
AutoChangeMusic \[ 308 \]
automaticBars \[ 594 \]
automatische Ausrichtung von Silben \[ 245 \]
automatische Beblattung \[ 77 \]
automatische Bunddiagramme \[ 357 \]
automatische Kombination von Stimmen \[ 166 \]
Automatische Versetzungszeichen \[ 24 \]
Automatischer Systemwechsel \[ 306 \]
automatischer Systemwechsel und relativ Modus \[ 307 \]
automatisches Aufteilen von Noten \[ 74 \]
B \[ 5 \]
Bezier-Kurven \[ 602 \]
Bögen, über Noten \[ 123 \]
Bögen, gleichzeitig \[ 123 \]
Bögen, gleichzeitige Phrasierung \[ 126 \]
Bögen, laissez vibrer \[ 50 \]
Bögen, manuelle Platzierung \[ 123 \]
Bögen, mehrfach \[ 123 \]
Bögen, Phrasierung \[ 123 \]
Bögen, unter Noten \[ 123 \]
Bögen, verändern \[ 602 \]
Backend \[ \text{509, 572, 576} \]
backslashed digits \[ 694 \]
\backslashed-digit \[ 694 \]
Balken in Kadenz \[ 69 \]
Balken in polymetrischer Notation \[ 71 \]
Balken mit Knie \[ 78 \]
Balken und Gesangstext \[ 80 \]
Balken und Melismen \[ 78 \]
Balken zwischen Systemen \[ 305 \]
Balken, automatisch \[ 77 \]
Balken, eigene Regeln \[ 77 \]
Balken, Einstellungen \[ 77 \]
Balken, gespreizt \[ 602 \]
Balken, letzter in Partitur \[ 86 \]
Balken, letzter in polyphonier Stimme \[ 86 \]
Balken, manuell \[ 88 \]
Balken, \texttt{partcombine} und \texttt{autoBeamOff} \[ 79 \]
Balken, Taktartstandard \[ 61 \]
Balken, Unterteilung \[ 84 \]
Balken, Zeilenumbrüche \[ 78 \]
Balkenpausen, mehraktig \[ 58 \]
Ballon \[ 211 \]
balloon-interface \[ 212 \]
Balloon_engraver \[ 211, 212 \]
\balloonGrobText \[ \text{211, 746} \]
\balloonLengthOff \[ 211 \]
Anhang E: LilyPond-Index 807

balloonTextMm ........................................... 211
\balloonText ........................................... 211, 746

BalloonTextItem ........................................... 212

banjo-c-tuning ........................................... 365
banjo-modal-tuning ........................................... 365
banjo-open-d-tuning ........................................... 365
banjo-open-dm-tuning ........................................... 365
Banjo-Stimmung ........................................... 365
Banjo-Tabulatur ........................................... 320
Banjo-Tabulature ........................................... 364
\bar ........................................... 92, 97, 746
Bar_engraver ........................................... 396
Bar_number_engraver ........................................... 101
barCheckSynchronize ........................................... 102
BaritoneSchlüssel ........................................... 16
BarLine ........................................... 97
BarNumber ........................................... 97, 101
\barNumberCheck ........................................... 102, 746
BarNumberVisibility ........................................... 97
Barre, anzeigen für bundierte Saiteninstrumente ........................................... 361
Barre, anzeigen für Bundinstrumente ........................................... 361
Barre, Gitarre ........................................... 339
Baroku-Pizzicato ........................................... 319
barType ........................................... 97
base_shortest_duration ........................................... 537
baseMoment ........................................... 80
BassFigure ........................................... 405, 407
BassFigureAlignment ........................................... 405, 407
BassFigureBracket ........................................... 405, 407
BassFigureContinuation ........................................... 405, 407
BassFigureLine ........................................... 405, 407
Bassnote in Akkorden ........................................... 392
Basso continuo ........................................... 402
Bassschlüssel ........................................... 16
bayati ........................................... 443
\beam ........................................... 79, 87, 91, 306, 335
\bar ........................................... 672
beamInterface ........................................... 79, 87, 91
Beam_ engraver ........................................... 80
BeamEvent ........................................... 79, 91
beamExceptions ........................................... 80, 746
BeamForbidEvent ........................................... 79, 87
\beatStructure ........................................... 80

Bebalken in taktloser Musik ........................................... 69
Bebalken in Kadenzten ........................................... 69
Bebalkung in polyrhythmerischer Notation ........................................... 71
Bebalkung nach Taktschlag ........................................... 85
Bebalkung, automatisch, Einstellungen ........................................... 80
Bebalkung, Taktrhythmen ........................................... 61
Beenden eines Systems ........................................... 181
Beenden eines Notensystems ........................................... 181
Beenden von Notenlinien ........................................... 181
Befehle zur Textausrichtung ........................................... 231
Beginn eines Notensystems ........................................... 174
Beginn von Wiederholung ........................................... 143
Beginnen eines Notensystems ........................................... 181
Beginnen von Notenlinien ........................................... 181
Beispiel aus dem Leben ........................................... 164, 166, 304
Beispiel der arabischen Musik ........................................... 444
\bendAfter ........................................... 128, 746
Beschriftung ........................................... 113
Beschriftung über mehrere Seiten ........................................... 236
Beschriftung über mehrfach ausgerichteten Text ........................................... 228
Beschriftung mit Bedingung ........................................... 464
Beschriftung, Blocksatz ........................................... 230
Beschriftung, mehrzeilig ........................................... 229
Beschriftung, Notation einfügen ........................................... 234
Beschriftung, Notationsobjekte einfügen ........................................... 224
Beschriftung, Sonderzeichen ........................................... 223
Beschriftung, Text ........................................... 230
Beschriftung, Zentrieren auf der Seite ........................................... 285
Beschriftungskonstruktionen in Scheme ........................................... 122, 123.

besondere Arpeggio-Symbole ........................................... 131
besondere Notenköpfe ........................................... 34
besondere Zeichen, Text ........................................... 224
Bezifferter Bass ........................................... 402
Bilder einbinden ........................................... 233
Bindebögen und Akkorde ........................................... 49
Bindebögen wiederholen ........................................... 49
Bindebögen, Aussehen ........................................... 50
Bindebögen, durchgehend ........................................... 50
Bindebögen, gepunktet ........................................... 50
Bindebögen, gestrichelt ........................................... 50
Bindebögen, verändern ........................................... 602
Bindebogen ........................................... 49
Bindebogen in alternativem Schluss ........................................... 140
Bindebogen in Wiederholung ........................................... 140
Bindebogen und Wiederholung ........................................... 143
Bindebogen, Gesangstext ........................................... 249
Binderand ........................................... 505
binding_offset ........................................... 505
Bison ........................................... 710
Blöcke, Text ........................................... 229
Blasinstrumente ........................................... 374
Blasinstrumente, Blocksatz, Text ........................................... 230
BNF ........................................... 710
Bogengruppe ........................................... 249
Bogen, Anzeige ........................................... 317
Bogen, halb durchgehend, halb gestrichelt ........................................... 126
Bogen, halb gestrichelt, halb durchgehend ........................................... 124
Bogen, Strichelung definieren ........................................... 124

\bold ........................................... 225, 647
\book ........................................... 448, 451
\bookOutputName ........................................... 450, 746
\bookOutputSuffix ........................................... 450, 746
\bookpart ........................................... 449, 451, 514
bookTitleMarkup ........................................... 461
bottom_margin ........................................... 501
box ........................................... 231, 647
brace ........................................... 178
bracket ........................................... 121, 178, 231, 310, 672
Bratschsenschlüssel ........................................... 16
break ........................................... 514
break-align-symbols ........................................... 599
break-alignment-interface ........................................... 726, 740
break-visibility ........................................... 590
breakable ........................................... 78
breakbefore ........................................... 458
breathe ........................................... 127, 746
Breathing_sign_ engraver ........................................... 128
BreathingEvent ........................................... 128
BreathingSign ........................................... 128, 424, 425
\breve ........................................... 41, 43, 52, 54
Bunddiagramm-Beschriftung .......................... 339
Bunddiagramme ........................................... 346
Bunddiagramme, eigene .................................. 346
Bunddiagramme, Flageolett ................................ 346
Bunddiagramme, Position .................................. 346
Bunddiagramme, gedämpfte Noten ........................ 346
bundierte Saiteninstrumente, Akkordformen .......... 350
bundierte Saiteninstrumente, Fingersatz der rechten Hand 350
bundierte Saiteninstrumente, Flageolett ................. 361
bundierte Saiteninstrumente, gedämpfte Noten ........ 361
bundierte Saiteninstrumente, Position und Barré anzeigen 361
bundierte Saiteninstrumente, Saitenstimmung .......... 335
Bundinstrumente, Akkordformen ......................... 350
Bundinstrumente, Fingersatz der rechten Hand ........ 359
Bundinstrumente, Flageolett .............................. 361
Bundinstrumente, gedämpfte Noten ....................... 361
Bundinstrumente, Position und Barré anzeigen ........ 361
Bundinstrumente, Saitenstimmung ....................... 335
Bundsteg .................................................... 505

C
C-Schlüssel .................................................. 16
cadenzarenge ............................................. 70, 111
\cadenzaoff ............................................... 69
\cadenzason .............................................. 69
casura .................................................... 426, 424
callback ................................................... 708
Callback functions ....................................... 597
Capo ......................................................... 342
\caps ...................................................... 647
\cavum .................................................... 426, 432
\center-align ............................................. 228, 656
\center-column ........................................... 229, 657
centering a column of text ................................ 657
\change ................................................... 305
changing direction of text columns ..................... 658
chords ..................................................... 290
\char ...................................................... 605
check-consistency ....................................... 504
ChoirStaff ................................................ 178, 180, 270
Chor-Tenorschlüssel ..................................... 17
chord ....................................................... 153, 388, 395
chord-Akkorde ........................................... 388
Chord_name_engraver .................................. 395
chordChanges ............................................. 395
\chordmode ............................................... 5, 13, 348
ChordName ............................................... 395
chordNameExceptions .................................. 398
chordNameLowerCaseMinor ............................... 396
ChordNames .............................................. 191, 348, 399, 395
chordNameSeparator .................................... 397
chordNoteName .......................................... 397
chordPrefixSpacer ....................................... 398
\chordRepeats ........................................... 324, 747
chordRootName .......................................... 396
Chords .................................................... 388, 390, 393, 395, 400, 402, 405, 407
\chords ................................................... 394
Chornoten ............................................... 279
Chorposition ............................................. 175
Christian Harmony-Notenkörpfe ......................... 37
church mode ............................................. 21
\circle ................................................... 231, 673
circling text ............................................. 673
Circular .................................................. 705
\clef ..................................................... 16, 415, 423, 434, 747
\clef ..................................................... 20
\clef-interface .......................................... 20
\clef-remover .......................................... 20
Clef .................................................... 157, 159
\clef ..................................................... 157
\clef-remover .......................................... 157
Cluster ................................................... 157
\cluster ................................................ 104, 113, 705
\compressFullBarRests ................................ 58, 747
\compressMMRests ...................................... 747
\concat .................................................. 658
concatenating text ..................................... 658
concert pitch ............................................ 24
\consists ............................................... 565
\context ............................................... 554, 560
\context in Vlayout-Umgebung .......................... 560
ContextChange .......................................... 306
Contextual ............................................... 522, 524, 528, 552
Contexts .................................................. 402
controlling general text alignment ..................... 660
\controlPitch ........................................... 484
Copyright-Zeichen ...................................... 117
\cresc .................................................. 116
creating a table ........................................ 702
creating empty text objects ............................ 697
creating horizontal spaces in text ..................... 662
creating text fractions ................................ 695
creating vertical spaces in text ....................... 660, 699
\cresc .................................................. 117
Crescendo ............................................... 116
crescendo ............................................... 121
crescendo-event ........................................ 197
Crescendo-Klammer ..................................... 116
Crescendiftakt .......................................... 116
Crescendoklammer ...................................... 595
Anhang E: LilyPond-Index 809

\crescHairpin .............................................. 117
\crescTextCresc .............................................. 117
cross .................................................. 34
\cross-staff ................................................. 308
crossStaff .................................................. 747
cue notes, removing ....................................... 202
cue-notes .................................................... 288
cueClef ...................................................... 198, 747
cueClefUnset ................................................. 747
cueDuring .................................................... 198, 747
cueDuringWithClef .......................................... 198, 747
CueVoice ..................................................... 198, 203, 288
currentBarNumber .......................................... 97, 111
custodes ..................................................... 413
custom ....................................................... 413
custom ....................................................... 411, 413

D
D'al Segno .................................................. 113
D.S. al Fine .................................................. 104
Dämpfung, bünderte Saiteninstrumente ................. 361
Dämpfung, Bundinstrumente ................................ 361
Dal Segno ..................................................... 104
Dateien einfügen ........................................... 476
Datenstruktur ............................................... 451
Dauer .......................................................... 41
Dauer, Standard ............................................. 42
Dauern skalieren .......................................... 48
Daumenbezeichnung ........................................ 113, 705
deaNote ..................................................... 747
\decal .................................................... 116
\decrease .................................................. 117
Decrescendo ............................................... 138
decrescendo ................................................ 121
default ...................................................... 24
default-staff-staff-spacing ................................ 521
Default_bar_line_engraver ................................ 74
defaultBarType .............................................. 97
\defaultTimeSignature .................................... 61
defineBarLine .............................................. 747
Definieren von eigenen Bunddiagrammen ............... 349
\demim.... ................................................ 426, 432
\denies ...................................................... 565, 566, 567
\descens .................................................. 426, 432
dicke der Notenlinien einstellen .......................... 181
didaktischer Versetzungzeichensatz ..................... 29
\dieAusgabe Verbessern .................................... 552, 596
\dim ...................................................... 117, 389
\dimHairpin ............................................... 117
Diminuendo .................................................. 116
\dimTextDecr ................................................. 117
\dimTextDecresc ............................................ 117
\dimTextDim ............................................... 117
\dir-column ............................................... 658
discant ...................................................... 689
Diskantsymbole, Akkordeon ................................ 311
\displayLilyMusic ......................................... 496, 747
displayMusic ............................................... 747
displayScheme .............................................. 747
Divisi, Gesangstext ....................................... 270
divisio ...................................................... 423, 424
\divisioMaier ............................................... 424
\divisioMaxima .............................................. 424
\divisioMinima .............................................. 424
divisionen .................................................. 423
dodecaphonic ............................................... 29
dodekaphoner Versetzungzeichensatz .................... 29
doit .......................................................... 129
doit .......................................................... 128
Doppel-B ..................................................... 5
Doppelpunktension ......................................... 42
Doppelschlag ............................................... 131
doppelte Taktartensymbole ................................ 71
doppelter Taktstrich ....................................... 92
dorian ....................................................... 20
Dorisch ...................................................... 20
DotColumn .................................................. 43
Dots .......................................................... 43
\downdownward .............................................. 42
\downeightarrow ............................................ 42
double flat ................................................... 7
double sharp .................................................. 7
Double_percent_repeat_engraver ......................... 149
doubleflat ................................................... 681
DoublePercentEvent ........................................ 149
DoublePercentRepeat .................................... 149
DoublePercentRepeatCounter ......................... 149
\doubleroot ................................................. 149
drawing a line across a page ................................ 674
drawing beams within text ................................ 672
drawing boxes with rounded corners .................... 676
drawing boxes with rounded corners around text ....... 679
drawing circles within text ................................ 673
drawing dashed lines within text ........................ 673
drawing dotted lines within text ......................... 674
drawing ellipse around text ................................ 675
drawing lines within text .................................. 674
drawing oval around text .................................... 677
drawing paths ............................................... 678
drawing solid boxes within text ......................... 676
drawing squiggled lines within text ..................... 675
drawing triangles within text ............................. 680
Drehten von Objekten ...................................... 595
Dreiklange ................................................... 389
Druckreihenfolge .......................................... 224
Drucken von Sonderzeichen ............................... 590
\drummode ................................................... 174
Drums ........................................................ 365
DrumStaff ................................................... 174, 175, 373
DrumVoice .................................................... 373
Dudelsack .................................................... 377
Dur ........................................................... 20
Anhang E: LilyPond-Index

Duration names notes and rests .................................. 43
durchgehender Legatobogen ........................................ 123
durchgezogenen Hals ................................................. 107
durchsichtiger, Objekte .............................................. 590
durchsichtige Noten ................................................... 207\dynamic .......................................................... 121, 618
dynamic-event ........................................................... 197\dynamicDown ........................................................... 118
DynamicLineSpanner .................................................. 118, 121\dynamicNeutral ......................................................... 118
Dynamics ............................................................... 121, 218
DynamicText ........................................................... 121\dynamicUp .............................................................. 118
Dynamik .................................................................. 116
Dynamic Kontext erzeugen an einer Note ...................... 554
Dynamik, vertikale Position ......................................... 118
Dynamik, zentriert für Tasteninstrumente ..................... 304
Dynamikzeichen, Anmerkung ........................................ 121
Dynamikzeichen, eigene .............................................. 121
Dynamikzeichen, Klammer ......................................... 121

E
\easyHeadsOff .......................................................... 35
\easyHeadsOn .......................................................... 35
Ebenen (layer) ......................................................... 590
Editorial annotations .................................................. 205, 207, 208, 209, 210, 211, 212, 214, 215
editorische Dynamikzeichen ....................................... 121
editorische Noten ...................................................... 210
eigene Balkenregeln .................................................... 77
eigene Bunddiagramme ............................................... 338, 343
Eigene Bunddiagramme definieren ................................ 349
eigene Dynamikzeichen .............................................. 121
eigene Kontexte erstellen ............................................ 554
Eigene Größeinstimmung, Tabulator ................................ 335
Eigene Tabulaturen .................................................... 335
Eigenschaften ........................................................... 572
Eigenschaften von Grob .............................................. 574
Eigenschaften, die Schnittstellen
besitzen können ....................................................... 709
ein System, Mehrstimmigkeit ...................................... 157
Ein zusätzliches System erscheint ................................ 138, 147, 568
Einbinden von Graphik ................................................. 233
Eine Funktion auf alle
Layout-Objekte anwenden .......................................... 555
Einfärben von Objekten .............................................. 208, 590
Einfärben von Stimmen .............................................. 161
einfügen von Dateien .................................................. 476
Einfügen von Notationsobjekten .................................... 234
einfache closure ........................................................ 708
Einfache Lieder setzen ............................................... 241, 242
Eingabe von Noten parallel .......................................... 171
Eingabezweck, Struktur ............................................... 451
eingebundene Graphik im Text .................................... 231
Einmal verändern von Kontexten ................................... 575
Einstellung von Hilfslinien .......................................... 181
einzelnes Notensystem ............................................... 174
Einzug .................................................................. 193\ellipse ................................................................. 675
Emmentaler font ....................................................... 625
encapsulated postscript output ...................................... 486
enclosing text in a box with rounded corners ................. 679
enclosing text within a box ........................................ 647
Ende von Wiederholung .............................................. 143\endSpamers ............................................................. 143
Engraver hinzufügen und entfernen ......................... 74, 75
Engraver, in Kontexte einfügen .................................. 566
Engravers and Performers ......................................... 552
Entfernen eines Stencil .............................................. 589
Entfernen von Stichnoten ........................................... 201
Entfernen von Stufen in Akkorden ............................. 392
Entfernen von Tönen aus Akkorden ............................ 391
Episena ............................................................... 425
Episena_engraver ...................................................... 425
EpisenaEvent ........................................................... 425
EPS-Ausgabe ........................................................... 486\epslfile ................................................................. 233, 676
Erinnerungsvorzeichen .............................................. 417
Erklärungsblase ....................................................... 211
erste Klammer ......................................................... 136
erweiterte Akkorde ................................................... 390\expressive ............................................................. 113, 117
Espressivo ............................................................... 117
Espressivo-Artikulation ............................................. 117
eventChords ........................................................... 747\expandFullBarRests ..................................................... 56, 58
explicitClefVisibility ................................................. 592
explicitKeySignatureVisibility .................................... 592
Expressive marks .................................................... 521
\extra_offset ........................................................... 521\eyeglasses .............................................................. 695

F
\ff ............................................................. 116
F-Schlüssel ........................................................... 16
Faßbarkeit, Mensuralnotation ..................................... 417
Färben von Objekten ................................................. 590
Färben von Stimmen .................................................. 161
Fülllinie ................................................................. 253
Füllung um Text ...................................................... 232
fall ................................................................. 129
falls ................................................................. 128
Farbe ................................................................. 208
Farbe, RGB ........................................................... 209
Farben Liste ........................................................... 623
farbige Noten ........................................................ 208
farbige Noten in Akkorden ........................................ 209\featherDurations ....................................................... 91, 748
fermata ................................................................. 113, 681
fermataMarkup ......................................................... 57, 58, 133
Fermate ................................................................. 104, 113, 705
Fermate über Mehraktspänen ..................................... 57
Fermate an Taktstrich ................................................. 219
Fetal font ............................................................... 625\fff ................................................................. 116\ffff ................................................................. 116\fffffff .............................................................. 116
fifth ................................................................. 5
figured bass ........................................................... 402, 413
FiguredBass ............................................................ 191, 405, 407
Figuren, Namen ..................................................... 283\fill-line ............................................................... 230, 659
\fill-with-pattern .................................................... 659
\filled-box ........................................................... 233, 676
gestopft .................................................. 113
gestrichelter Legatobogen .......................... 123
gestrichelter Phrasierungsoben .................... 126
geteilte Stimmen ....................................... 281
getrennter Gesangstext ............................... 270
Gitarren-Akkordnotation ............................ 75
Gitarrengriffsymbol .................................. 338
Gitarrennotenköpfe ................................... 34
Gitarrenschlagrhythmus, Notation ................. 75
Gitarrentabulatur ....................................... 320
Gitterlinien ............................................. 212
gleizzeitige Bögen .................................... 123
gleizzeitige Noten: Versetzungszeichen .......... 30
gleizzeitige Phrasierungsoben ...................... 126
Gleiten in Tabulaturen ................................ 326
Gleiten nach oben/unten .............................. 128
Glissando .................................................. 129, 130, 589
\glissando .............................................. 129, 130
Glissando, Akkorde .................................... 326
Glissando, nach oben .................................. 128
Glissando, nach unten ................................ 128
Glissando, unbestimmt ................................ 128
glyph ...................................................... 708
Glyphe ...................................................... 708
Größe der Schriftart .................................. 225
Größe von Notensystem verändern .................. 214
Größe von Objekten .................................... 188
Größe, Papier ........................................... 498
Größe, Seite ............................................. 498
\grace ..................................................... 105, 748
grace notes .............................................. 109
Grace_beam_engraver ................................ 109
Grace_engraver ......................................... 109
Grace_spacing_engraver .............................. 109
GraceMusic .............................................. 109
Grammatik von LilyPond ............................. 710
grand staff .............................................. 178
GrandStaff .............................................. 30, 178
GraphicalObjectInterfaces .......................... 708
graphical objects ...................................... 708
Graphik einbinden ..................................... 233
Graphik, eingecluded .................................. 231
Graphische Notation .................................. 233
graphische Objekte ................................... 708
graphische Objekte, Eigenschaften .................. 574
graphische Objekte, Schmittstellen ............... 709
Gregorianische quadratische Neumenligaturen . 425
Gregorianischer Choral, Transcription .......... 174
GregorianTranscriptionStaff ....................... 174, 175
grid-line-interface .................................... 214
grid-point-interface ................................... 214
Grid_line_span_engraver ............................. 212, 214
Grid_point_engraver .................................. 212, 214
gridInterval .......................................... 212
GridLine ................................................. 214
GridPoint ............................................... 214
Griff Fingersatz ....................................... 205
Griffsymbole, bundierte Saiteninstrumente ...... 338
Griffsymbole, Bundinstrumente ..................... 338
grob ....................................................... 708
Grob ....................................................... 569
Grob-Eigenschaften .................................... 574
grob-interface .......................................... 569, 570, 709, 711
grobdescriptions ....................................... 748
Grobe, Sichtbarkeit .................................... 589
Grobs, verändern ..................................... 590
grow-direction ......................................... 91
Grundlegende Optionen auf der ................... 91
Kommandozeile für LilyPond .......................... 486
Grundton eines Akkordes ............................. 391
Grundton eines Akkords ................................ 389
I
Ich höre Stimmen .................................. 161, 367
Ictus ............................................. 705
Illustrationen im Text .......................... 231
immutable-Eigenschaften ..................... 709
immutable-Objekte ............................. 709
importing stencils into text .................. 699
Improvisation ................................. 40
\improvisationoff .............................. 40, 75
\improvisationon .............................. 40, 75
in ............................................. 583
in markierte Noten einfügen .................. 479
incipit ......................................... 748
\inclinatum .................................. 426, 432
\include ....................................... 476
include-settings ................................ 482
indent ......................................... 193, 505, 541
inherit-acceptability ......................... 749
inlining an Encapsulated PostScript image ........................................ 476
inner-margin .................................. 676
inserting music into text ...................... 684
inserting PostScript directly into text ........ 678
inserting URL links into text ................. 680
\inStaff\Segno .................................. 139, 749
instrument-specific-markup-interface ........ 387, 697
Instrumentbezeichnungen ...................... 488
Instrumente, transponierende ................. 10
Instrumentenbezeichnung, komplexe ........ 192
Instrumentenbezeichnung, zentriert ........ 192
Instrumentenbezeichnungen .................... 191
Instrumentenbezeichnungen zu anderen
  Kontexten hinzufügen ......................... 193
Instrumentenbezeichnungen, wechseln ........ 193
Instrumentengruppe ........................... 175
Instrumentenwechsel .......................... 194
InstrumentName ................................ 195
InstrumentSwitch ............................. 288
\instrumentSwitch .............................. 194, 749
interface ...................................... 709
Internals Reference ........................... 552
interval ....................................... 5
\inversion ..................................... 13, 749
\ionian ....................................... 20
Ionisch ........................................ 20
iraq ............................................ 443
\italianChords ................................ 398
\italic ....................................... 225, 649
item-interface ................................ 570

J
Justierung von Notensystemen ................. 181
\justified-lines ................................ 230, 701
justify ........................................ 230, 664
\justify-field .................................. 663
\justify-line ................................... 663
\justify-string ................................ 664
justifying lines of text ....................... 701
justifying text ................................ 664

K
künstliches Flageolett .......................... 318
Kadenz .......................................... 69
Kadenz und Seitenumbrüche .................... 70
Kadenz und Seitenumbruch ..................... 69
Kadenz und Zeilenumbrüche .................... 70
Kadenz und Zeilenumbruch ..................... 69
Kadenz, Ausrichten an ......................... 110
Kadenz, Bekalkung .............................. 69
Kapo ............................................ 342
Kasten, Graphik ................................ 231
\keepWithTag ................................... 479, 749
\key ............................................. 20, 38, 749
key-signature ................................... 419, 423
key-cancellation-interface ..................... 21
key-signature-interface ....................... 21
Key_ engraver ................................... 21
Key_performer ................................... 21
Keyboards ....................................... 304, 306, 308, 310, 311, 315
KeyCancellation ............................... 21
KeyChangeEvent .................................. 21
KeySignature ..................................... 21, 419, 423, 443
kievan notation ................................ 433, 434
KievanStaff ....................................... 433
KievanVoice ...................................... 433
skillCues ....................................... 202, 749
Kirchenpausen .................................. 58
Kirchentonarten ................................. 20
Klammer, Crescendo ............................ 116
Klammer, erste (Wiederholung) ............... 136
Klammer, geschweift ........................... 175
Klammer, vertikal .............................. 175
Klammer, Wiederholung ......................... 143
Klammer, Wiederholung mit Text ............. 144
Klammer-Arpeggio über Systeme .............. 133
Klammern ....................................... 214
Klammern um Noten ............................. 210
Klammern um Vorzeichen ...................... 6
Klammern, Analyse ............................. 214
Klammern, Crescendo, schräg ................. 595
Klammern, Graphik ............................. 231
Klammern, spitze ................................ 152
Klammern, unterschiedliche Größen .......... 237
Klammern, Verschachteln ..................... 179
Klang .......................................... 487
Klavier, Pedalbezeichnung ...................... 310
Klavier-Versetzungszeichen .................... 28
Klaviersystem .................................. 304
Klaviersystem, zentrierte Dynamik .......... 304
kleinere Noten .................................. 198
Knall-Pizzicato ................................ 319
Kollisionen, vertikal, vermeiden .............. 535
Klimpern von Stimmen ........................................ 166
Komma-Intervalle ........................................... 445
Komprimieren von Noten .................................... 48
Kontext, Layoutreihenfolge .................................. 567
Kontexte erstellen ........................................... 554, 555
Kontexte und Engraver ....................................... 159, 552
Kontexte, am Leben erhalten ............................... 555
Kontexte, einmal verändern ................................ 575
Kontexte, Lebensdauer ....................................... 555
Kontexte, neue definieren ................................... 565
Kontexteigenschaften verändern ............................ 565
Kontextveränderungen rückgängig machen ............. 575
Kontextveränderungen, Einstellungen ändern .......... 575
Kreuz .......................................................... 9
Kreuzpunkte und tweak ...................................... 578
Kreuzpunkte, Bézier-Kurven ................................. 602
Kreis .......................................................... 14
Kreuzungsköpfe .............................................. 34
kurz ......................................................... 443
kurze Instrumentenbezeichnungen .......................... 191

L

Länge und Dicke von Objekten .............................. 188, 583
Länge von Zeilen .............................................. 541
\label ......................................................... 749
Laissez vibrer ................................................ 50
laissez viber ................................................. 52
\laissezVibrer ............................................... 50
\laissezVibreTie ............................................ 52
\laissezVibertieColumn ................................... 52
landscape ..................................................... 749
language ...................................................... 749
languageRestore ............................................ 749
languageSaveAndChange .................................. 749
\large ....................................................... 204, 227, 649
\larger ..................................................... 225, 227, 649
last-bottom-spacing ....................................... 503
Lautstärke .................................................... 116
layer (Ebenen) .............................................. 590
\layout ...................................................... 451, 509, 560
\layout file ................................................ 511
layout objects .............................................. 708
Layout, Partitur ............................................. 509
Layout-Schnittstelle .................................... 509
Layoutobjekte .............................................. 708
ledger line .................................................... 184
ledger-line-spanner-interface ............................ 35
LedgerLineSpanner ......................................... 35
leere Systeme verstecken .................................. 188
Leerzeichen .................................................. 453
Leerzeichen, Gesangstext .................................. 242, 249
left aligning text .......................................... 665
\left-align .................................................. 228, 665
\left-brace ................................................ 696
\left-column ............................................... 665
left-margin ................................................. 504
Legatobögen ............................................... 123
Legatobögen, manuelle Platzierung ....................... 123
Legatobögen, verändern ................................... 602
Legatobögen zur Phrasierung ............................... 126
Legatobogen, gepunktet ................................... 123
Legatobogen, gestrichelt ................................ 123
Legatobogen, massiv ....................................... 123
Legatobogen-Stil .......................................... 123
lexer .......................................................... 709
\leheal ..................................................... 113
Lieder ........................................................ 243, 278
ligature ........................................................ 411, 412, 421, 432
Ligaturen ..................................................... 412
Ligaturen der quadratischen Neumennotation .......... 425
Ligaturen, weiße Mensuralnotation ....................... 420
ligatures in text ............................................ 658
lilypond aufrufen .......................................... 485
\LilyPond grammar ........................................ 710
line .......................................................... 184, 665
line-spanner-interface .................................. 589
line-width ................................................... 504, 541
linea .......................................................... 426, 432
\LineBreakEvent ............................................ 514
\lineprall .................................................... 113
Linien zwischen Systemen ................................ 212
Linien, Gitter .............................................. 212
Liste der Farben ............................................ 623
Liste der vorhandenen Schriftarten ....................... 239
Literatur ..................................................... 396, 400
locrian ....................................................... 20
Locrisch ..................................................... 20
\longa ....................................................... 41, 43, 52, 54
Longa-Pause ................................................ 52
\longfermata ............................................... 113
\lookup ..................................................... 696
\lower ....................................................... 228, 666
lowering text .............................................. 666
\ltoc ......................................................... 113
\ly:add-context-mod ..................................... 759
\ly:add-file-name-alist ................................ 759
\ly:add-interface ......................................... 759
\ly:add-listener .......................................... 759
\ly:add-option ........................................... 759
\ly:all-grob-interfaces ................................ 759
\ly:all-options ........................................... 759
\ly:all-stencil-expressions ................................ 759
\ly:angle .................................................. 759
\ly:assoc-get ............................................. 759
\ly:axis-group-interface:add-element .................. 759
\ly:basic-progress ...................................... 760
\ly:beam-score-count .................................. 760
\ly:book-add-bookpart! .................................. 760
\ly:book-add-score! ..................................... 760
\ly:book-book-parts ..................................... 760
\ly:book-header .......................................... 760
\ly:book-paper ........................................... 760
\ly:book-process ......................................... 760
\ly:book-process-to-systems ................................ 760
\ly:book-scores .......................................... 760
\ly:book-set-header! ..................................... 760
\ly:book? .................................................. 760
\ly:box? ................................................... 760
\ly:bp ...................................................... 760
\ly:bracket ............................................... 760
\ly:brodcast .............................................. 760
\ly:camel-case-xlisp-identifier ......................... 760
\ly:chain-assoc-get ..................................... 761
\ly:check-expected-warnings ............................ 761
\ly:cm ..................................................... 761

Anhang E: LilyPond-Index
Mehristimmigkeit ........................................ 157
Mehristimmigkeit, ein System .......................... 157
 Mehrtaktpause mit Fermate .......................... 57
 Mehrtaktpausen ....................................... 53, 56
 Mehrtaktpausen und Fingersatz ...................... 60
 Mehrtaktpausen, ausschreiben ....................... 56
 Mehrtaktpausen, Beschriftung ......................... 57
 Mehrtaktpausen, komprimieren ....................... 56
 Mehrtaktpausen, Positionierung ...................... 58
 Mehrtaktpausen, Text hinzufügen .................... 57
 mehrzeiliger Text ...................................... 229
\melsisma .......................................... 250, 253
Melsima ........................................... 250, 253
\melsismaEnd ....................................... 250
 Melismen, Balken .................................... 78
 Melodiehymne: Anzeige ................................ 75
 Melodieinterpretation, Krebs ......................... 14
 Melodieinterpretation, Umkehrung .................. 13
 Melodieinterpretationen, modal ...................... 14
 Melodieposition, modal ................................ 14
 Melodieumkehrung, modal ............................ 15
 Mensur .............................................. 415
 mensural notation ..................................... 411, 412, 414, 415, 416, 417, 418, 419
 Mensuraligaturen ..................................... 420
 Mensuralmusik, Transkription ......................... 78
 Mensuralnotation ..................................... 410
 MensuralStaff ........................................ 174, 175, 414
 Mensuralstil .......................................... 411
 MensuralVoice ........................................ 414
 Mensuralstriche ...................................... 178
 merging text ......................................... 657, 666
 meter ................................................ 73
 Metronombezeichnung .................................. 65
 Metronome ........................................... 68
 metronome mark ....................................... 68
 MetronomeMark ....................................... 68
 metronomic indication .................................. 68
 Metrum ............................................... 60
 Metrum, Noten ohne .................................... 69, 111
 Metrum, polytisch ..................................... 71
 Mezzosopranschlüssel .................................. 16
 \mf .................................................. 116
 \midi ............................................... 451
 MIDl ............................................... 23, 487
 MIDI und Wiederholungen ............................ 491
 MIDI, Akkordsymbole ................................ 490
 MIDI, Artikulationen .................................. 490
 MIDI, Artikulier-Skript ................................ 490
 MIDI, Mikrotoni ....................................... 490
 MIDI, Rhythmen ...................................... 490
 MIDI, Tonhöhen ....................................... 490
 MIDI, Vierteltöne ..................................... 490
 MIDI-Instrumentenbezeichnungen ..................... 622
 MIDI-Übertragung ..................................... 489
 MIDI-Übertragung ..................................... 23
 MIDI-Umgabe ......................................... 489
 Mikrotoni ............................................ 5, 8
 Mikrotoni in MIDI ..................................... 490
 min-systems-per-page ................................ 506
 minimum-Y-extent .................................... 521

m ...................................................... 389
\magnify ............................................. 225, 649
 magnifying text ....................................... 649
 magifyMusic ......................................... 749
 magifyStaff .......................................... 749
 magstep .............................................. 204, 583
 maj ..................................................... 389
 \major ............................................... 20
 major seven symbols .................................. 398
 majorSevenSymbol ................................... 396
 makam .............................................. 445, 446
 makamlar ............................................. 440, 445, 446
 make-dynamic-script .................................. 121
 make-pango-font-tree ................................ 239
 \makeDefaultStringTuning ........................... 157, 749
 Manuals .............................................. .1
 manuelle Balken ..................................... 88
 manuelle Balken, Richtung zuweisen ............... 89
 manuelle Balken, Verzierungen ....................... 89
 manuelle Systemwechsel ................................ 305
 manuelle Taktstriche .................................. 93
 manuelle Wiederholungszeichen .................... 143
 manuelles Überschriftezeichen ....................... 103
 \map-markup-commands ................................ 701
 Maqam .............................................. 440
 maqam .............................................. 443
 Marcato ............................................. 113, 705
 \marcato ............................................ 113
 \mark .................................................. 102, 219, 749
 Mark_engraver ...................................... 105, 221
 \markalphabet ....................................... 697
 Markle ............................................... 479
 MarkEvent .......................................... 105, 221
 Markieren von Abschnitten ......................... 102
 markierte Noten behalten ............................ 479
 markierte Noten entfernen ......................... 479
 \markletter ......................................... 697
 markup .............................................. 219, 222, 223
 markup, Syntax ...................................... 223
 markup-markup-spacing ................................ 503
 \markusystem-spacing ................................ 503
 \markuplist ......................................... 222, 236, 237
 markupMap .......................................... 749
 massiver Legatobogen ................................ 123
 Matrize (stencil) ..................................... 711
 Matrize, entfernen .................................... 589
 mac-systems-per-page ................................ 506
 \maxima ............................................. 41, 43, 52, 54
 Maxima-Pause ........................................ 52
 measureLength ........................................ 80, 111
 measurePosition ...................................... 68, 111
 Medicaea, Editio ..................................... 410, 411
 \medium ............................................ 650
 Mehr Information ..................................... 161, 440, 477, 486, 487, 490, 491, 570, 596
 mehrere Dynamikzeichen an einer Note ............. 117
 Mehrere Noten auf einmal ............................ 166
 mehrere Phrasierungsbögen ......................... 126
 mehrere Stimmen ..................................... 161
 mehrfache Bögen ..................................... 123
 mehrmator Bögen ..................................... 109
 mehrseitiger Text ..................................... 236
Anhang E: LilyPond-Index

819

\begin{tabular}{ll}
minimumFret & 323, 358 \\
minimumPageTurnLength & 516 \\
minimumRepeatLengthForPageTurn & 516 \\
\texttt{\textbackslash minor} & 20 \\
\texttt{\textbackslash minorChordModifier} & 398 \\
mirroring markup & 679 \\
mixed & 310 \\
\texttt{\textbackslash mixolydian} & 20 \\
mixolydisch & 20 \\
\texttt{n} & 583 \\
modele Transformationen & 14 \\
modale Transpositionen & 14 \\
modale Umkehrung & 15 \\
\texttt{\textbackslash modallInversion} & 15, 750 \\
\texttt{\textbackslash modallTranspose} & 14, 750 \\
modern & 26 \\
modern-cautionary & 27 \\
modern-voice & 27 \\
modern-voice-cautionary & 27 \\
modern-Warnung-Versetzungszeichenstil & 26 \\
moderne Versetzungszeichen & 27 \\
Moderner Stil, Versetzungszeichen & 26 \\
moderner Tabulatur-Schlüssel & 337 \\
moderners Versetzungszeichenstil & 26, 27 \\
modernes Versetzungszeichenstil mit Warnungen & 27 \\
moderner Versetzungszeichenstil mit Warnungen & 27 \\
Warnungen für Stimmen & 27 \\
modern-tab & 377 \\
Modi, in Akkorden & 389 \\
Modifikatoren, Akkorde & 389 \\
Modus & 20 \\
Moll & 20 \\
Mordent & 113, 705 \\
\texttt{\textbackslash mordent} & 113 \\
\texttt{\textbackslash mp} & 116 \\
\texttt{\textbackslash multi-measure-rest} & 60 \\
\texttt{\textbackslash MultiMeasureRest} & 60 \\
\texttt{\textbackslash MultiMeasureRestNumber} & 60 \\
\texttt{\textbackslash MultiMeasureRestText} & 57, 60 \\
\texttt{\textbackslash Music classes} & 198 \\
\texttt{\textbackslash Musica ficta} & 419 \\
\texttt{\textbackslash musicglyph} & 104, 681 \\
\texttt{\textbackslash musicMap} & 750 \\
Musk komprimieren & 48 \\
Musik ohne Metrum, Seitenumbrüche & 70 \\
Musik ohne Metrum, Umbrüche & 69 \\
Musik ohne Metrum, Zeilenumbrüche & 70 \\
Musikalische Ausdrücke erklärt & 448 \\
Musikalische Ausdrücke & 35, 37, 40, 197 \\
in einander verschachteln & 185, 188, 568 \\
Musikalische Funktionen & 606, 606 \\
Musikalische Funktionen darstellen & 497, 577, 578 \\
Musikanalyse & 214 \\
Musikbachtäfelchen & 104 \\
Musikobjekte, Einfügen & 234 \\
musikwissenschaftliche Analyse & 214 \\
mutable-Objekte & 709 \\
\texttt{\textbackslash N} & 44 \\
N-töne & 43 \\
N.C.-Symbol & 394 \\
Nachschlag & 106 \\
\texttt{\textbackslash name} & 565 \\
Name von Sängern & 272 \\
Namens von Figuren & 283 \\
Naming conventions of objects & 709 \\
and properties & 682 \\
\texttt{\textbackslash natural} & 28 \\
\texttt{\textbackslash neo-modern} & 28 \\
\texttt{\textbackslash neo-modern-cautionary} & 28 \\
\texttt{\textbackslash neo-modern-cautionary-Versetzungszeichenstil} & 28 \\
\texttt{\textbackslash neo-modern-voice} & 29 \\
\texttt{\textbackslash neo-modern-voice-cautionary} & 29 \\
\texttt{\textbackslash neo-moderner Versetzungszeichenstil} & 28 \\
\texttt{\textbackslash neo-moderner Versetzungszeichenstil pro Stimme} & 29 \\
\texttt{\textbackslash neo-moderne Versetzungszeichenstil pro Stimme mit Warnungen} & 29 \\
Neomensuralstil & 411 \\
\texttt{\textbackslash neue Definitionen von} & 237 \\
Beschriftungsmöglichkeiten & 121 \\
neue Dynamikzeichen & 554 \\
niederes Notensystem & 174 \\
\texttt{\textbackslash new} & 554 \\
\texttt{\textbackslash New_fingering_engraver} & 207, 569 \\
nicht metrische Musik, Umbrüche & 69 \\
Nicht-ASCH-Zeichen & 482 \\
Nicht-Textschriftarten in Beschriftung & 237 \\
nichtmusikalische Symbole & 233 \\
niente, al & 119 \\
\texttt{\textbackslash no-reset} & 30 \\
\texttt{\textbackslash noBeam} & 89 \\
\texttt{\textbackslash noBreak} & 514 \\
\texttt{\textbackslash NonMusicalPaperColumn.} & 538 \\
\texttt{\textbackslash nonstaff-nonstaff-spacing} & 521 \\
\texttt{\textbackslash nonstaff-relatedstaff-spacing} & 521 \\
\texttt{\textbackslash nonstaff-unrelatedstaff-spacing} & 521 \\
\texttt{\textbackslash noPageBreak} & 515, 750 \\
\texttt{\textbackslash noPageTurn} & 516, 750 \\
\texttt{\textbackslash normal-size-sub} & 650 \\
\texttt{\textbackslash normal-size-super} & 226, 650 \\
\texttt{\textbackslash normal-text} & 650 \\
\texttt{\textbackslash normale Wiederholung} & 136 \\
\texttt{\textbackslash normaleize} & 204, 227, 651 \\
Notation für Streicher & 316 \\
Notation innerhalb von Beschreibung & 235 \\
Notation innerhalb von Text & 235 \\
Notation, Aiken & 37 \\
Notation, Erklärungen & 211 \\
Notation, graphische & 233 \\
Notationobjekte, Einfügen & 234 \\
\texttt{\textbackslash note} & 682 \\
\texttt{\textbackslash note head} & 417, 434 \\
\texttt{\textbackslash note value} & 682 \\
\texttt{\textbackslash note-by-number} & 730, 733, 735 \\
\texttt{\textbackslash note-collision-interface} & 35, 37, 40, 197 \\
\texttt{\textbackslash note-head-interface} & 35, 37, 40 \\
\texttt{\textbackslash Note_head_line_engraver} & 308 \\
\texttt{\textbackslash Note_heads_engraver} & 35, 37, 40, 74 \
\end{tabular}
Anhang E: LilyPond-Index

Note\_spacing\_engraver \hfill 208

NoteCollision \hfill 166

NoteColumn \hfill 166

Noten \hfill 35, 37, 40

Noten ausdehnen \hfill 181

Noten in Klammern \hfill 210

Noten komprimieren \hfill 181

Noten ohne Metrum \hfill 111

Noten ohne Takt \hfill 69, 111

Noten verschmelzen \hfill 161

Noten verstecken \hfill 207

Noten wiederholt schreiben \hfill 145

Noten zu Akkorden verbinden \hfill 153

Noten, Aufteilen \hfill 74

Noten, doppelpunktuiert \hfill 42

Noten, durchsichtig \hfill 207

Noten, farbig \hfill 208

Noten, farbig in Akkorden \hfill 209

Noten, kleiner \hfill 198

Noten, paralato \hfill 34

Noten, punktiert \hfill 42

Noten, Schriftgröße \hfill 204

Noten, Stichnoten \hfill 198

Noten, transponieren \hfill 10

Noten, unsichtbar \hfill 207

Noten, Wechsel zwischen Systemen \hfill 205

Noten-Schriftzeichen \hfill 104

Notenabstände, Abschnitte definieren \hfill 538

Notenabstände, horizontal \hfill 538

Notenbezeichnungen, arabisch \hfill 441

Notenbezeichnungen, Deutsch \hfill 5

Notenbezeichnungen, Holländisch \hfill 5

Notenbezeichnungen, Standard \hfill 5

Oberste Ebene, Text \hfill 222

Notes and interfaces \hfill 709

Objekte und Schnittstellen \hfill 472, 709

Objekte verändern \hfill 709

Objekte, Drehen \hfill 595

Objekte, einfärben \hfill 590

Objekte, farbig \hfill 208

Objekte, Graphik im Text \hfill 231

Objekte, Sichtbarkeit \hfill 589

Octavation \hfill 22

OctaveCheck \hfill 9, 750

Offset \hfill 705

Offen \hfill 113

Offene Saiten, anzeigen \hfill 317

On \hfill 800

Oktavbestimmung, relativ \hfill 2

Oktaveprüfung \hfill 500

Oktavenmodus (relativ) und Akkorde \hfill 4

Oktavierte Schlüssel, Sichtbarkeit \hfill 594

Oktavierung \hfill 22

Oktavierungskorrektur \hfill 9

Oktavtransposition \hfill 17

Oktavwechsel: Tonhöhe \hfill 2

Omit \hfill 750

On-the-fly \hfill 461, 697

Once \hfill 574, 575, 750

OneVoice \hfill 157

Open \hfill 113, 317

Optimieren \hfill 576

Optimierungsmethoden \hfill 47, 576, 578
Optischer Ausgleich .......................... 537, 538
Oratorium ........................................ 279
Orchester, Streicher ......................... 316
Orgelpedal-Bezeichnung ................. 113
Orgelpedalabzeichnung .................. 705
\oriscus .................................. 426, 432
Ornament ......................................... 113
Ormente .......................................... 105
Osmanische Musik ......................... 445
Ossia .............................................. 184
ossia ........................................... 188, 189
Ossia-Systeme ................................. 184
ottava ......................................... 22, 750
ottava-bracket-interface .............. 22
Ottava_spanner_ engraver .............. 22
OttavaBracket .................................. 22
outer-margin .................................. 505
output-def ....................................... 790
outside-staff-horizontal-padding ...... 535
outside-staff-padding .................... 535
\oval ........................................... 677
\overlay ......................................... 666
\override ...................................... 574, 697
Override im Gesangstextmodus .......... 242
\override in \lyricmode .................. 242
override rückgängig machen ........... 575
override, nur einmal ..................... 575
\override-lines .................................. 572
overrideProperty ................................ 750
\overrideTimeSignatureSettings ... 61, 751
overriding properties within text markup .. 697
\overtie .......................................... 651
overtie-ing text ............................... 651

P

\p ............................................. 116
\pad-around ..................................... 232, 666
\pad-markup .................................... 232, 666
\pad-to-box ..................................... 232, 667
\pad-x ........................................... 232, 667
padding text .................................. 666
padding text horizontally .............. 667
page-breaking ................................. 507
page-breaking-system-system-spacing .. 507
page-count ....................................... 507
\page-link ..................................... 698
\page-ref ........................................ 698
\page-ref ........................................ 474
\page-spacing-weight ....................... 508
\pageBreak ...................................... 515, 751
\pageTurn ....................................... 516, 751
palmMute ....................................... 751
palmMuteOn ..................................... 751
Pango .......................................... 237
\paper ........................................... 451, 499
paper-height .................................... 500
paper-width ..................................... 504
Papier, Ausrichtung ....................... 499
Papier, quer ..................................... 499
Papierformat ...................................... 499
Papiergröße ...................................... 498
Parallele Notation, Eingabe ................ 171
\parallelMusic .................................. 171, 751
parentheses-interface ................... 210
ParenthesesItem ............................... 210
Parenthesis_ engraver ..................... 210
\parenthesize .................................. 210, 677, 751
parlato .......................................... 289
parlato-Notenköpfe .......................... 34
Parsesan font ..................................... 625
parser ............................................. 710
part .............................................. 170
\partcombine ................................... 166, 751
\partcombineApart ............................. 167
\partcombineAutomatic ....................... 167
\partcombineChords .......................... 167
partcombineDown ................................ 751
partcombineForce ................................ 751
PartCombineMusic ................................ 170
\partcombineSoloI ............................ 167
\partcombineSoloII ............................ 167
\partcombineUnisono .......................... 167
partcombineOp ................................. 751
\partial ........................................... 68, 136, 138, 751
partieller Takt ................................ 68
Partitur ......................................... 175
Partitur, Layout ............................... 175
Partituren und Stimmen .................. 477
\path ............................................. 678
paths, drawing .................................. 678
\pattern ......................................... 698
Pausen .......................................... 52
Pausen verschieben, automatisch ...... 161
Pausen, Aufteilen ............................. 74
Pausen, Ganzakt ................................ 56
Pausen, ganzaktig ............................. 53
Pausen, Kirchenstil .......................... 58
Pausen, mehrere Takte ausschreiben ..... 56
Pausen, mehrere Takte komprimieren ..... 56
Pausen, Mehrtakt ................................ 56
Pausen, mehraktig ............................. 53
Pausen, Mensuralnotation ................. 418
Pausen, unsichtbar ............................ 54
Pausen, vertikale Position festlegen ..... 53
Pausen, Zusammenhalten ................... 60
Pausen, Zusammenstöße .................... 60
Pausendauern .................................... 52
Pausenzeichen ................................... 127
Pedal, Harfe ..................................... 315
Pedal, sustenuto ............................... 310
Pedal-Bezeichnung ............................ 113
Pedalbezeichnung ............................. 310
Pedalbezeichnung, Klammer ............. 310
Pedalbezeichnung, Stile .................... 310
Pedalbezeichnung, Text ..................... 310
Pedaldiagramme, Harfe .................... 315
\pedalSustainStyle ............................ 310
percent .......................................... 147
percent repeat .................................... 149
Percent_repeat_ engraver ................ 149
PercentRepeat .................................. 149
PercentRepeatCounter ...................... 149
PercentRepeatMusic .......................... 149
Percussion ...................................... 365, 366, 367, 373
Percussionsnotensystem ................... 174
Pizzicato, Bartók
Pizzicato, Knall-
Platz zwischen Notensystemen
Platz innerhalb von Systemgruppen
Pitches
Phrasierungszeichen
PhrasingSlur
\phrasingSlurDashed
\phrasingSlurDashPattern
\phrasingSlurDotted
\phrasingSlurDown
\phrasingSlurHalfDashed
\phrasingSlurHalfSolid
\phrasingSlurNeutral
\phrasingSlurSolid
\phrasingSlurUp
Phrygian
Piano
Piano, Pedalbezeichnung
Piano cautioned
Piano-System
Piano-Versatzzeichensstil
PianoPedalBrackets
PianoStaff
Pitch
Pitch names
Pitch_Sharp, engraver
\pitchedTrill
Pitches
\prallup
Praller
Prallermordent
Pralle
PropertyRevert
PropertySet
PropertyTweak
PropertyUnset
Psalms
Pizzicato, Bartók
Pizzicato, Knall-
placement horizontal brackets around text
placement parentheses around text
placement vertical brackets around text
Platz innerhalb von Systemgruppen
Platz um Text
Platzierung, Layouteinstellungen
PNG-Ausgabe
pointAndClickOff
pointAndClickOn
pointAndClickTypes
\polymetric
\polymetric time signature
Polytonale Notation und Balken
Polymetrische Partitur
Polyphonie
Polyphonie, ein System
Polyphony
Portato
R

R............................. 52
rückgängig machen von Kontextveränderungen........ 575
ragged-bottom............... 501
ragged-last.................. 501
ragged-last-bottom......... 501
ragged-right................ 501
Rahmen, Text................ 231
railroad tracks............... 128
\raise ........................ 228, 667
raising text.................. 667
Rallantando in MIDI.......... 490
Rand um Text................. 232
Rand, überhängender Text... 217
rast........................... 443
Ratschlauch, Editio ......... 411
rautenförmige Notenköpfe.... 318
Rautennotenköpfe.............. 34
rechte Hand, Fingersatz für bundierte
   Saiteinstrumente............. 359
rechte Hand, Fingersatz für Bundinstrumente....... 359
reduceChords.................. 753
referencing page labels in text 701
referencing page numbers in text 698
Referenz der Interna......... 552
regelmäßige Zeilenumbrüche..... 513
Regeln zur Benennung von Objekten
   und Eigenschaften........... 709
RehearsalMark................ 105, 221
reine Container, Scheme...... 603
Relative...................... 2
\relative ........................ 2, 5, 13, 307, 753
Relative Oktavbestimmung..... 2
relative Tonlänge, Akkorde.... 153
RelativeOctaveCheck.......... 10
RelativeOctaveMusic.......... 5
relativer Modus und Akkorde.... 4
relativer Modus und automatischer
   Systemwechsel................. 307
Relativ Oktavenmodus und Transposition............ 5
religious music................ 290
RemoveAllEmptyStaves........ 756
\RemoveEmptyStaves......... 188, 189, 756
\removeWithTag.............. 479, 753
removing cue notes............ 202
Renaissancemusik............... 178
\repeat ......................... 136, 142
\repeat percent............... 147
\repeat tremolo............ 149
\repeatCommands............. 143
RepeatedMusic............... 142, 145, 147
Repeats........................ 142, 145, 147, 149, 151
RepeatSlash.................. 149
RepeatSlashEvent............. 149
\repeatTie.................... 49, 140, 267
repetitive Musik.............. 145
\replace ..................... 652
resetRelativeOctave......... 753
\rest .......................... 52, 683
Rest................................ 54
\rest-by-number.............. 683
rest-event..................... 197
Rest_ engraver................ 74
RestCollision................ 106
rests or multi-measure-rests within text by
   log and dot-count.............. 683
rests or multi-measure-rests
   within text by string......... 683
\retrograde................... 14, 753
\reversturn ................. 113
\revert....................... 575
RevertProperty ............... 572
\revertTimeSignatureSettings .. 63, 753
\rfz .......................... 116
rgb-color..................... 209
RGB-Farbe...................... 209
\rh ............................ 113
Rhythmen in MIDI.............. 490
RhythmicStaff................. 40, 77, 174, 175
Rhythmische Zeilenumbrüche..... 43
rhythmisches Notensystem..... 174
Rhythms 43, 47, 49, 52, 54, 55, 60, 65, 69, 70, 73, 74
   77, 79, 87, 91, 92, 97, 101, 102, 105, 109, 111, 112
Rhythmus der Melodie anzeigen................. 75
richtung von Notenhälsen..... 210
right aligning text........... 668
\right-align .................. 228, 668
\right-brace .................. 698
\right-column ................. 668
\right-margin ................ 504
\rightHandFinger............. 359, 753
\roman .......................... 652
\rotate ........................ 668
rotating text................ 668
\rounded-box ................. 231, 679
\rtoe ........................ 113
R .................................. 56

S

S.............................. 54
Sängername.................... 272
Satz, mehrere................. 48
Sackpfeife..................... 377
sacred harp-Notenköpfe........ 37
\sacredHarpHeads ............. 37
\sacredHarpHeadsMinor ....... 38
Saite, offen.................. 317
Saitenführung für Bundinstrumente..... 335
Saitenzahl..................... 320
\sans.......................... 652
SATB.......................... 279
Satzzeichen................... 242
scalable vector graphics output 486
\scale .......................... 679
\scaleDuration.................. 48, 71, 753
scaling markup................ 679
scaling text.................. 669
Schachtedlung von Systemen... 179
Scheme objekt................. 710
Scheme, reine Container....... 603
Scheme, unreine Container.... 603
Scheme-Tutorium.............. 552
Schlüssel....................... 5, 16
Schlüssel Alter Musik......... 16
Schlüssel, C.................... 16
Schlüssel, F.................... 16
Schüsse, greg. Choral.......... 422
Schüsse, G...................... 16
Schlüssel, Kiever Notation .......................... 433
Schlüssel, Mensuralnotation .......................... 414
Schlüssel, modern, Tabulatur .......................... 337
Schlüssel, Sichtbarkeit der Oktavierung .............. 594
Schlüssel, Sichtbarkeit nach 
expliziter Änderung .................................. 592
Schlüssel, transponierend ............................... 17
Schlaggruppen ........................................... 85
Schlagrhythmus, Gitarre ................................. 75
Schlagzeuge .............................................. 365, 367
schließenende Taktstriche ............................... 92
Schluss, alternativer in Wiederholung ................. 136
Schmittstelle von graphischen Objekten ............. 709
Schmittstelle, Layout ................................... 569
Schnittstellen für Programmierer ...................... 596
Schottischer Dudelsack ................................. 377

die sicheren Halbklappen .............................. 40
Schriftart verändern .................................... 225
Schriftarten, für das gesamte Dokument ändern .... 239
Schriftarten, Hintergrundinformation ................... 237
Schriftarten, Liste zum Auswählen .................... 239
Schriftarten, Nicht-Text in Beschriftung ............. 237
Schriftarten, vorhandene auflisten .................... 239
Schriftartenfamilien, definieren ....................... 239
Schriftfamilie ........................................... 708
Schriftfamilien .......................................... 226
Schriftgröße ............................................. 225
Schriftgröße (Notation) ändern ....................... 204
Schriftgröße (Notation), Standard ..................... 204
Schriftgröße, Einstellung .............................. 511
Schriftschnitt verändern ............................... 225
Schriftschnitte ......................................... 226
Schriftzeichen, Notenschrift ........................... 104
scordatura .............................................. 21
\score .................................................. 112, 712, 715
\score .................................................. 447, 451, 684
\score ist ein (einziger) zusammengesetzter 
musikalischer Ausdruck .............................. 448
\score-linen ........................................... 702
score-markup-spacing ................................ 503
score-system-spacing ................................ 503
scoreTitleMarkup ..................................... 461
Script ................................................... 113, 115, 425
Script_engraver ....................................... 425
ScriptEvent ............................................ 425
Seconda volta .......................................... 136
Segue ................................................. 104, 113, 705
segno .................................................. 94, 113
Segue an Taktstrich .................................. 219
Segmente ............................................... 498
Segmentformat ......................................... 499
Seitenanfang, überhängender Text ..................... 217
Seitenumbrüche ........................................ 514, 541
Seitenumbrüche in Kadenzen ......................... 69, 70
Seitenumbrüche in Musik ohne Metrum ............... 70
Seitenumbrüche in nicht metrischer Musik ........... 69
Seitenzahlen, automatische Numerierung .......... 507
Seitenzahlen, unterdrücken ........................... 507
self-alignment-interface ............................. 570, 596
self-alignment-X ...................................... 521
self-alignment-X ...................................... 444
Semi-Form .............................................. 443
Semicirculus ........................................... 705
semflat ............................................... 685
\semiflat .............................................. 398
\semisharp ............................................ 685
separater Text ........................................ 222
Septakorde ............................................ 389
sesqui-B ............................................... 8
sesqui-Kreuz .......................................... 8
\sesquiflat ............................................ 685
\sesquisharp .......................................... 685
\set ................................................. 80, 572
set-octavation ........................................ 22
setting extent of text objects ....................... 701
setting horizontal text alignment ..................... 661
setting subscript in standard font size ............. 650
setting superscript in standard font size .......... 650
settingsFrom .......................................... 753
Setzen von Sonderzeichen ......................... 224
Setzen von Text ....................................... 223
\sf .................................................... 116
\sff ................................................... 116
\sfz ................................................... 7, 685
shape ................................................ 753
sharp .................................................. 414
\shiftIf ............................................... 161
\shiftOn ............................................... 161
\shiftOff ............................................. 161
\shiftOmn ............................................. 161
\shiftOmn ............................................. 161
\short.indent ......................................... 193, 506
\shortfermata ........................................ 113
show-available-fonts ................................ 239
showFirstLength ...................................... 485
showKeySignature .................................... 377
showLastLength ....................................... 485
\showStaffSwitch ..................................... 308
Sichtbarkeit und Farbe von Objekten ............... 55, 191,
Sichtbarkeit von Objekten ........................... 589
Sichtbarkeit von oaktvierten Schlüsseln .......... 594
side-position-interface ............................. 578, 596
Séquenzen, erste definieren ......................... 507
sigmoid congruentiae ................................ 705
\sigmoidcongruentiae ................................ 113
sikah ............................................... 443
Silben spreizen ....................................... 260
silver ............................................... 511
\simple .............................................. 652
simple text strings ................................... 652
simple text strings with tie characters .......... 685
simultane Noten und Versetzungszeichen ........... 30
Simultaneous notes .................................. 153, 157, 161, 166, 170, 173
single .............................................. 753
skalierbare Vektographik-Ausgabe ................. 486
Skalieren von Dauern ................................ 48
Skip .................................................. 54
\skipp ............................................... 54, 266, 753
SkipMusic ............................................ 55
skipTypesetting ...................................... 485
\slash-repeat_engraver ................................ 149
slashChordSeparator ................................ 397
slashed digits ........................................ 699
\slashed-digit ........................................ 699
\slashedGrace ......................................... 105, 753
Slide in Tabulature .................................. 326
Slur ................................................... 125, 133
slur ................................................ 125
\slur ................................................ 125
slur-event ........................................ 197
\slurDashed ...................................... 123
\slurDashPattern ............................... 124, 753
\slurDotted ..................................... 123
\slurDown ........................................ 123
\slurHalfDashed ................................ 124
\slurHalfSolid .................................. 124
\slurNeutral ..................................... 123
\slurSolid ........................................ 123
\slurUp ........................................... 124
\small .......................... 204, 227, 653
\smallCaps ....................................... 653
\smaller ........................................ 225, 227, 653
smob .............................................. 710
\snappizzicato .................................. 113
Solesmes .......................................... 411
solo-Stellen ..................................... 166
Sonderzeichen ................................... 482
Sonderzeichen in Textbeschrifungen .......... 224
Sopranschlüssel ................................... 16
Sopranschlüssel in C ............................ 16
sos ................................................... 310
sostenuto-Pedal .................................. 310
SostenutoEvent .................................. 311
\sostenutoiff ..................................... 310
\sostenutoOn ..................................... 310
SostenutoPedal ................................... 310
SostenutoPedalLineSpanner ...................... 311
Southern-Harmony-Notenköpfe .................. 37
\southernHarmonyHeadsMinor ................. 38
\app .............................................. 116
Spacing .......... 501, 503, 506, 508, 511, 512, 514, 515, 516, 517, 518, 520, 525, 527, 528, 535, 536, 538, 540, 541, 548, 550, 551
spacing ........................................... 537
spacing-spanner-interface ..................... 739, 741
SpacingSpanner ................................... 538
spacingTweaks ..................................... 754
SpanBeam ......................................... 308
SpanBar ........................................... 97
spitze Klammern .................................. 152
\app .............................................. 116
Sprache, Tonhöhenbezeichnung in anderer ... 7
Sprechgesang ..................................... 289
Spreizen von Silben ................................ 260
Springen zwischen Systemen .................... 305
Stücke durch Variablen organisieren ..... 173, 477, 482
Staccatissimo ...................................... 113
\staccatissimo .................................... 113
\staccato ......................................... 113, 115
\smu .............................................. 657
Staff ... 30, 33, 74, 97, 175, 178, 191, 195, 215, 712
staff ............................................. 175, 184, 188
Staff notation .................................... 68, 175, 178, 180, 181, 184, 188, 191, 195, 198, 203
Staff symbol, erstellen .......................... 181
staff-affinity ..................................... 521
staff-staff-spacing .............................. 521
\staff-symbol-interface ....................... 184
Staff.midInstrument ................................ 488
Staff_symbol_engraver .......................... 188, 191
StaffGroup ....................................... 101, 178, 180
staffgroup-staff-spacing ...................... 521
StaffGrouper ..................................... 281, 522, 524, 527, 538
StaffSpacing ..................................... 538
StaffSymbol ...................................... 175, 184, 188
Standard Notendauer ............................ 41
Standard-Schriftgröße (Notation) .......... 204
Standard-Versetzungszeichenstil .......... 24, 26
Standardkontexteigenschaften, ändern .... 560
Standardnotenbezeichnungen .................. 42
Standardnotendauer ............................. 61
Standardtafelinstellungen ..................... 97
Standardtaktstrich, Änderung ................ 278
\start-repeat ................................. 143
\startStaff ....................................... 181, 184
\startTrillSpan ................................ 133
staves .......................................... 175
\stdBass ......................................... 690
\stdBassIV ....................................... 691
\stdBassV ......................................... 692
\stdBassVI ........................................ 693
\stem .............................................. 210, 211, 308, 310
stem-interface ................................... 211
stem-spacing-correction ...................... 537
stem-engraver ................................... 91, 211
\stemDown ......................................... 89
\stemNeutral ..................................... 210
Stempel (stencil), entfernen .................. 589
\stemRightBeamCount ......................... 89
\stemUp ........................................... 210
\stencil .......................................... 699, 711
\stencil .......................................... 589
Stichnoten ........................................ 195, 198, 285
Stichnoten innerhalb von rhythmischer Kombination .......... 47
Stichnoten, entfernen ............................ 201
Stichnoten, Formatierung ...................... 198
Stil von Übungszeichen ....................... 103
Stil von Legatobögen ........................... 123
Stil von Taktangaben ........................... 61
Stile, Notenköpfe ............................... 34, 646
Stile, Stimmen .................................... 161
Stimme ........................................... 157
Stimmfolgen ...................................... 308
Stimmen, Stile ................................. 161
Stimmen, Versetzungsarten .................... 27
Stimmen, Versetzungszeichenstil .......... 26
Stimmen kombinieren ........................... 159, 161
Stimmen, Versetzungszeichenstil mit ... 20
Warnung für Stimmen ......................... 27
Stimmen, zitieren ................................ 195
Stimmfolgestriche ............................... 308
Stimmgruppe ..................................... 175
Stimmkreuzung .................................. 308
Stimmumfang ..................................... 31
Stimmung ......................................... 365
\stopGroup ....................................... 214
Anhang E: LilyPond-Index

\topped ........................................ 113
\topStaff ...................................... 181, 184, 188
\topTrillSpan ................................ 133
\storePredefinedDiagram .................. 350, 754
Strecker, Text ................................ 217
Streicher, Text-, Formatierung ............ 217
Streicher ...................................... 316
Streichquartett ................................ 317
Striche zur Stimmverfolgung ............... 308
Striche: Notenköpfe ........................... 40
Strichnotenköpfe .............................. 40
\StringNumber ................................ 322
\stringTuning ................................. 336, 754
\stropha ...................................... 335, 346
Strofe ......................................... 360
\sub ............................................ 226, 653
Subasschlüssel ................................ 16
\subdivideBeams ............................... 84
\substring ..................................... 653
\subtraction .................................. 391
\suggestAccidentials ......................... 419
\sup ............................................ 226, 654
\super .......................................... 654
\sus ............................................ 392
SustainEvent .................................. 311
\sustainOff .................................... 310
\sustainOn ..................................... 310
SustainPedal .................................. 311
SustainPedalLineSpanner .................... 311
SVG-Ausgabe .................................. 486
\symbole ................................ ...... 219
\symbole, Akkord- ............................. 393
\symbole, Akkordeon ......................... 393
\symbole, nicht musikalische ............... 233
Synchronisation von Verzierungen .......... 109
System querende Häse ........................ 308
System, beenden ................................ 181
System, Chor .................................. 175
System, geschachtelt ......................... 179
System, Größe verändern .................... 184
\system-count ................................ 506
\system-separator-markup ................... 508
\system-system-spacing ....................... 503
System-Trennzeichen .......................... 180
\SystemBeginBegrenzer, geschachtelt .... 179
Systeme verdecken ............................ 188
Systeme, leere ................................ 188
Systeme, mehrere .............................. 175
Systeme, Tremolo zwischen ................ 151
Systeme, Zusammenstöße beim .......... 305
Systemwechsel ................................ 305
Systemwechsel von Stimmen ................ 308
Systemwechsel, automatisch ............... 306
Systemwechsel, manuell ..................... 305

\SystemStartSquare ......................... 179, 180
Systemwechsel von Stimmen ................ 308
Systemwechsel, automatisch ............... 306
Systemwechsel, manuell ..................... 305

\Tag ............................................. 479, 754
\tag ............................................. 479, 754
\tabChordRepetition ......................... 754
\tabChordRepeats ............................. 324, 754
\tabFullNotation ............................. 323
\table ......................................... 702
\table-of-contents ........................... 476, 702
\TabNoteHead ................................. 335
\TabStaff ...................................... 174, 175, 322, 335
Tabulatur ..................................... 174, 320
Tabulatur und Flageolett .................... 326
Tabulatur, Banjo ................................ 335, 364
Tabulatur, BaBsitarre ....................... 335
Tabulatur, Bratsche .......................... 335
Tabulatur, Cello ................................ 335
Tabulatur, eigene Saitenstimmung ......... 336
Tabulatur, Geige ................................ 335
Tabulatur, Gitarre ............................ 335
Tabulatur, Grundlegendes .................... 322
Tabulatur, Kontrabass ....................... 335
Tabulatur, Mandoline ....................... 335
Tabulatur, moderner Schlüssel ............. 337
Tabulatur, Saitenstimmung .................. 335
Tabulatur, Ukulele ............................ 335
Tabulaturen und Gleiten ..................... 326
Tabulaturen, eigen ........................... 335
Tabulatursystem ................................ 174
\TabVoice ..................................... 322, 335
Tag ............................................. 754
\tagGroup ...................................... 479, 754
Takt ............................................. 85
Takt, Noten ohne ................................ 111
Taktüberprüfung ................................ 102
Taktangabe .................................... 60
Taktangabe, Sichtbarkeit .................... 60
Taktangaben-Stil .............................. 61
Taktart, Mensuralnotation .................... 415
Taktart, Noten ohne ........................... 69
Taktart, Standardeigenschaften wiederherstellen 63
Taktart, Standardeinstellung ............... 61
Taktarten, arabisch ............................ 443
Taktarten, mehrere in Partitur ............. 559
Taktarten, polyrhythmisch ................... 71
Taktarten, unterschiedliche per System ... 559
Taktarten, zusammengesetzt ................. 73
Taktartensymbole, doppelt ................... 71
Taktartensymbole, unterteilt .......... 71
Takte verkürzen ................................ 68
Taktgruppen ................................... 85
Taktlinie, manuell ............................. 93
Taktlinie, Wiederholung ..................... 143
Taktlinien ..................................... 92
Taktlinien, Ausrichtung ..................... 100
Taktlinien, unsichtbar ....................... 92
Taktlose Musik, Bembalkung ............... 69

türkische Notenbezeichnungen .................. 445
Türkische Musik ................................ 445
Tab_note_heads_ engraver .................... 338
Tag ............................................. 754
\tag ............................................. 479, 754
\table ......................................... 702
\table-of-contents ........................... 476, 702
\TabNoteHead ................................. 335
\TabStaff ...................................... 174, 175, 322, 335
Tabulatur ..................................... 174, 320
Tabulatur und Flageolett .................... 326
Tabulatur, Banjo ................................ 335, 364
Tabulatur, BaBsitarre ....................... 335
Tabulatur, Bratsche .......................... 335
Tabulatur, Cello ................................ 335
Tabulatur, eigene Saitenstimmung ......... 336
Tabulatur, Geige ................................ 335
Tabulatur, Gitarre ............................ 335
Tabulatur, Grundlegendes .................... 322
Tabulatur, Kontrabass ....................... 335
Tabulatur, Mandoline ....................... 335
Tabulatur, moderner Schlüssel ............. 337
Tabulatur, Saitenstimmung .................. 335
Tabulatur, Ukulele ............................ 335
Tabulaturen und Gleiten ..................... 326
Tabulaturen, eigen ........................... 335
Tabulatursystem ................................ 174
\TabVoice ..................................... 322, 335
Tag ............................................. 479, 754
\tag ............................................. 479, 754
Takt ............................................. 85
Takt, Noten ohne ................................ 111
Taktüberprüfung ................................ 102
Taktangabe .................................... 60
Taktangabe, Sichtbarkeit .................... 60
Taktangaben-Stil .............................. 61
Taktart, Mensuralnotation .................... 415
Taktart, Noten ohne ........................... 69
Taktart, Standardeigenschaften wiederherstellen 63
Taktart, Standardeinstellung ............... 61
Taktarten, arabisch ............................ 443
Taktarten, mehrere in Partitur ............. 559
Taktarten, polyrhythmisch ................... 71
Taktarten, unterschiedliche per System ... 559
Taktarten, zusammengesetzt ................. 73
Taktartensymbole, doppelt ................... 71
Taktartensymbole, unterteilt .......... 71
Takte verkürzen ................................ 68
Taktgruppen ................................... 85
Taktlinie, manuell ............................. 93
Taktlinie, Wiederholung ..................... 143
Taktlinien ..................................... 92
Taktlinien, Ausrichtung ..................... 100
Taktlinien, unsichtbar ....................... 92
Taktlose Musik, Bembalkung ............... 69
Anhang E: LilyPond-Index 827

Taktnummer ........................................... 111
Taktnummer, Form ........................................ 98
Taktnummern ............................................. 97
Taktnummern, Zusammenstöße ......................... 101
Taktposition und Wiederholung ..................... 143
Taktschläge gruppieren ............................... 85
Taktstrich, doppelt .................................... 92
Taktstrich, Symbole anfügen ......................... 219
Taktstriche .............................................. 92
Taktstriche, Änderung von Standard ............... 97
Taktstriche, manuell .................................. 93
Taktstriche, schließend ................................ 92
Taktstriche, unsichtbar ................................ 62
Taktstriche, unterdrücken ............................ 594
Taktweise Wiederholungen ........................... 147
Taktzahlen ............................................. 97
Taktzahlen in Wiederholungen ....................... 99
Taktzahlen mit Buchstaben .......................... 99
Taktzahlen, gleichmäßige Abstände ............... 97
\taqasim ................................................. 377
taqasim .................................................. 443, 444
Tasteninstrumente, Notensystem .................. 304
Tasteninstrumente, zentrierte Dynamik .......... 304
teaching .................................................. 29
teaching-Versetzungszeichenstil ..................... 29
Tentakt .................................................. 68
\tempo ..................................................... 65
\tempo indication ...................................... 68
Tempo ..................................................... 65
Tempoanzeige ........................................... 65
Tempoanzeige innerhalb von N-tolen-Klammer ... 47
temporary ............................................... 754
Tenorschlüssel ........................................... 16
Tenorschlüssel, Chor .................................. 17
Tenuto ..................................................... 113, 705
\tenuto .................................................. 113, 115
\text .................................................... 217, 219, 221, 223, 225, 227, 231, 234, 236, 237, 239, 243
text ........................................................ 310, 654
Text über mehrere Seiten ............................ 236
Text über Mehrakttpausen ............................ 57
Text alleine ............................................ 222
Text am Taktschritt .................................. 219
Text an einer Melodie ausrichten ................ 245, 253
Text außerhalb des Randes .......................... 217
Text auf der Seite zentrieren ........................ 230
text columns, left-aligned .......................... 665
text columns, right-aligned ......................... 668
Text einrahmen ........................................ 231
Text in Volta-Klammer ............................... 234
Text mit Sonderzeichen ................................ 224
Text und Balken ...................................... 80
Text verziern .......................................... 231
Text, an Melodie ausgerichtet ...................... 244
Text, andere Sprachen ................................ 216
Text, Ausrichtung ................................... 228
Text, Blocksatz ....................................... 230
Text, horizontale Ausrichtung ..................... 228
Text, innerhalb des Randes behalten ............. 217
Text, mehrere Zeilen ................................ 229
Text, Notation innerhalb ............................ 235
Text, oberste Ebene ................................... 223
Text, Rand außen ..................................... 232
Text, Syntax ........................................... 223
Text, vertikale Ausrichtung ........................ 228
text-interface ......................................... 578, 697
text-script-interface ................................. 570
Textarten ............................................... 216
textartige Zeichen .................................... 219
Textausrichtung, Befehle ............................ 231
Textausrichtungsbefehle ............................ 231
Textbeschriftung ...................................... 223
Textbeschriftung über mehrere Seiten .......... 236
Textbeschriftung ausrichten ........................ 228
Textbeschriftung, Blocksatz ....................... 230
Textbeschriftung, mehrzeilig ....................... 229
Textbeschriftung, Notationsobjekte einfügen ... 234
Textbeschriftung, Sonderzeichen ................. 224
Textbeschriftungs-Ausdrücke ....................... 223
Textblocke .............................................. 229
Textblöcke ............................................. 211
Textelemente, nicht leer ............................ 216
Textgröße .............................................. 225
\textLengthDiff ......................................... 58, 217
\textLengthMin ......................................... 58, 217\TextScript .............................................. 115, 217, 223, 227, 231, 234, 236, 237, 387
TextSpanner ............................................ 219, 589
textSpannerDown ...................................... 218
textSpannerNeutral ................................... 218
textSpannerUp ......................................... 218
Textstrecke ............................................ 217
Textstrekker, Formatierung ......................... 217
Textzeichen ............................................ 219
The Emmentaler font ................................ 682
\thumb ..................................................... 113, 205
tie .......................................................... 52, 74, 654
tie ......................................................... 52
tie-ing text ............................................. 654
tieColumn ............................................... 52, 603
\tiesed-lyric ............................................ 685
\tieDashed ............................................... 50
\tieDashPattern ........................................ 754
\tieBotted ............................................... 50
\tieDown .................................................. 50
tiegerstellt ............................................ 226
\tiesNeutral ............................................. 50
ties, placement ........................................ 50
\tiesSolid ................................................ 50
\tieUp ..................................................... 50
\time ....................................................... 60, 80, 754
\timeSignature ......................................... 65
\times ..................................................... 43, 71, 754
\timesScaleddMusic ..................................... 47
\timesSignature ......................................... 65, 74
\timesSignatureFraction .............................. 71
\TimingTranslator ...................................... 65, 69, 74, 97, 112, 715
\time ...................................................... 204, 227, 655
\tocItem .................................................. 476, 754
Tonart ..................................................... 4, 20
Tonart, Mensuralnotation ............................ 419
Tonart, Sichtbarkeit nach expliziter Änderung ... 592
Tonarten, Gregor, Choral ............................ 423
Tonhöhe: Wechsel der Oktave ....................... 2
Tonhöhen in MIDI ...................................... 490
V

Varcode 113, 705
\Varcode \ Varcode 113
Variable 452
Variablen, Benutzung 477
Variablen, Gesangstext 254
Vaticana, Edito 410, 411
VaticanaStaff 174, 175, 421
VaticanaVoice 421
Vcenter 669
Veränderbare (mutable) Objekte 709
Verändern der Schriftart 225
Verändern der Schriftgröße 511
Verändern der Systemgrößen 511
Verändern von automatischer Behaltung 80
Verändern von Eigenschaften 572
verändern von Objekten 590
veränderte Akkorde 390
Veränderung des Notensystems 583
Veränderung von Kontexten nur einmal 575
Veränderung von Verzierungsnoten 107
\verbatimfile 699
Vermeidung von vertikalen Zusammenstößen 535
Verschachtelte Musik 171
verschachtelte Systemklammern 179
verschachtelte Wiederholung 143
Verschachtelung von Systemen 179
Verschieben von Noten 161
Verschieben von Objekten 228, 231
Verschieben von Pausen, automatisch 161
Verschiebung 571
Verschiebung von Gesangstext 255
Verschmelzen von Noten 161
Verschwinden von leeren Systemen 188
Versetzungszeichen 5
Versetzungszeichen an übergebundener Note 6
Versetzungszeichen für Klavier 28
Versetzungszeichen in Akkorden 30
Versetzungszeichen pro Stimme 27
Versetzungszeichen und gleichzeitige Noten 30
Versetzungszeichen, automatisch 24
Versetzungszeichen, Deutsch 5
Versetzungszeichen, Erinnerung 6
Versetzungszeichen, für Triller 135
Versetzungszeichen, greg. Choral 423
Versetzungszeichen, Kiever Notation 434
Versetzungszeichen, Mensuralnotation 419
Versetzungszeichen, moderne Stile 26
Versetzungszeichen, moderner Stil
mit Warnungen 27
Versetzungszeichen, musica ficta 419
Versetzungszeichen, piano cautionary 28
Versetzungszeichen, Standard 24
Versetzungszeichen, Viertelton 5
Versetzungszeichen, Viertelton 7
Versetzungszeichen, Warnung 6
Versetzungszeichenstil 24
Versetzungszeichenstil, Klover mit Warnungen 28
Versetzungszeichenstil, modern 26
Versetzungszeichenstil, modern-modern
mit Warnungen 28
Versetzungszeichenstil teaching 29
Versetzungszeichenstil Vergessen 30
Versetzungszeichenstil, modern... 27
Versetzungszeichenstil, modern mit
Warnung für Stimmen 27
Versetzungszeichenstil, modern-cautionary 26
Versetzungszeichenstil, neo-modern 28
Versetzungszeichenstil, neo-modern-voice 29
Versetzungszeichenstil, neo-modern-voice-cautionary 29
Versetzungszeichenstil, no reset 30
Versetzungszeichenstil, piano 28
Versetzungszeichenstil, Standard 26
Versetzungszeichenstil, Stimme 26
Versetzungszeichenstil, Zwölftonmusik 29
Versetzungszeichenstil: nicht zurücksetzen 30
Verstecken von Noten 207
Verstecken von Rhythmus-Systemen 189
Verstecken von Systemen 188
Verstecken von Systemen der Alten Musik 189
versteckte Notensysteme 184
\VerticalAxisGroup 191, 281, 521, 522, 524, 525,
527, 528, 756
vertically centering text 669
vertikale Ausrichtung von Text 228
vertikale Linien zwischen Systemen 212
vertikale Position von Dynamik 118
vertikale Positionierung 520
vertikale Zusammenstöße, vermeiden 535
Verwaltung der Zeiteinheiten 111
verwendbare Schriftarten auflisten 239
\verylongfermata 113
Verzierung innerhalb von rhythmischer Kombination 47
Verzierung innerhalb von Triolen 47
Verzierung, danach 106
Verzierungen 105
Verzierungen in MIDI 490
Verzierungen verändern 107
Verzierungen, Ausseln verändern 107
Verzierungen, manuelle Behandlung 89
Verzierungen, Synchronisation 109
Verzierungsnoten und Gesangstext 273
viele Stimmen 161
\VierstimmigeSATBPartitur 279
Vierteltöne 5
Vierteltöne in MIDI 490
Vierteltonversetzungszeichen 7
\\virga 426, 432
\\virgula 424
Vocal music 242, 278, 283, 288, 290
\Voice 33, 40, 157, 170, 198, 203, 249, 571
Voice enthält Noten 47, 161, 166
\VoiceStile 161
\Voice-Versetzungszeichenstil 26
\VoiceFollower 308, 589
\\voiceFourStyle 161
\\voiceNeutralStyle 161
\\voiceOne 157
\\voiceOne \\voiceFour 157
\\voiceOneStyle 161
\voices 755
\\voiceThreeStyle 161
\\voiceTwoStyle 161
\\void 496, 756
Vokalensemble ............................ 257, 260, 279, 297, 300
volta ....................................... 142
Volta ......................................... 136
Volta und Überbindung .................... 49
Volta-Klammer mit Text .................... 144
Volta-Klammern und Wiederholungen .... 49
Volta_ engraver ................................ 396
VoltaBracket ................................. 142, 145
VoltaKlammer, ändern ..................... 143
VoltaRepeatedMusic .......................... 142, 145
Vorhalt ...................................... 105
vorhandene Schriftarten auflisten .......... 239
Vorlage, arabische Musik .................. 444
Vorschlag ................................... 105
Vorschlag, mehrere Noten ................ 109
Vorzeichen .................................. 20
Vorzeichen in Klammern .................... 6
Vorzeichen, Erinnerung ..................... 6
Vorzeichen, greg. Choral ................... 423
Vorzeichen, Mensuralnotation .............. 419
Vorzeichen, Vierteltöne .................... 5
\vspace ..................................... 669

W
Walker-Formnotenköpfe ...................... 37
\wakerHeads ................................ 37
\wakerHeadsMinor .......................... 38
Warnungssatzzeichensymbole für Klavier 28
Warnungssatzzeichensymbole, neo-modern 28
Warnungsvorzeichen ......................... 6
Was sind Engraver? .......................... 74
Wechsel der Oktave ........................... 2
Wechsel des Systems, automatisch .......... 306
Wechsel des Systems, manuell .............. 305
Wechsel von Instrument ........................ 194
Wechsel zwischen Systemen ................. 308
Wechseln von Instrumententbezeichnungen 193
Weiße Mensuralligaturen ................... 420
weit aufeinander liegende Ballen .......... 78
whichBar ....................................... 97
\whitewout .................................. 700
Wie eine LilyPond-Eingabe-Datei funktioniert 453, 459
Wiederherstellen von
Taktart-Standardeigenschaften ............ 63
wiederholte Musik ........................... 145
Wiederholung mit alternativem Schluss ... 136
Wiederholung mit Auftakt ................... 138
Wiederholung mit \q .................................. 154, 324
Wiederholung und Bindebögen ............ 49
Wiederholung und Bindebogen ............. 143
Wiederholung und Zählzeit ................. 143
Wiederholung von Gesangstext bei
alternativen Endungen .................... 266
Wiederholung, alternative Schlüsse ....... 143
Wiederholung, aufklappen ................. 145
Wiederholung, Beginn ....................... 143
Wiederholung, Ende ......................... 143
Wiederholung, kurz ......................... 147
Wiederholung, manuell ...................... 143
Wiederholung, mehrdeutig .................. 143
Wiederholung, Prozent ...................... 147
Wiederholung, taktweise .................... 147
Wiederholung, Tremolo ........................ 149
Wiederholung, verschachtelt ............... 143
Wiederholung, VoltaKlammer .............. 143
Wiederholungen ................................ 95, 136
Wiederholungen in MIDI .......................... 491
Wiederholungen mit Überbindung ........ 140
Wiederholungen und Gesangstext .......... 262
Wiederholungen, alternative
Taktnummerierung .......................... 141
Wiederholungen, ausgeschrieben .......... 145
Wiederholungen, Taktnummer mit Buchstaben 141
Wiederholungsklammer mit Text ........... 144
Wiederholungstaktlinie ..................... 143
Wiederholungszeichen ....................... 92
Winds ......................................... 375, 377, 378, 379, 387
wirkliche Tonhöhe ............................ 5
\with ........................................ 558, 563
\with-color ................................... 208, 700
\with-dimensions ......................... 701
\with-dimensions-from ..................... 701
\with-link .................................. 701
\with-outline ............................... 701
\with-url ................................... 680
within-staff (Objekte innerhalb
 des Notensystems) ......................... 582, 583
\withMusicProperty ......................... 756
\woodwind-diagram ......................... 689
\wordwrap ................................... 236, 671
\wordwrap-field ............................. 670
\wordwrap-internal ......................... 703
\wordwrap-lines ................................ 236, 703
\wordwrap-string ........................... 671
\wordwrap-string-internal ................. 703
World music .................................. 441, 442, 443, 444, 445

X
x11-color .................................. 208, 209
x11-Farbe ................................... 209
X-offset ...................................... 521
X11-Farben ................................... 208
xNote ......................................... 756

Z
Zählzeit und Wiederholung .................. 143
Zahl der Notenlinien einstellen .......... 181
Zahl eines Taktes ............................ 97
Zahl von Saiten .............................. 320
Zeichen ....................................... 113
Zeichen, Übung: Formatierung ............ 103
Zeichen, textartige ......................... 219
Zeichnen im Text ............................. 231
Zeilenlänge .................................. 541
Zeilenumbrüche .............................. 92, 512
Zeilenumbrüche in Intervallen ............ 513
Zeilenumbrüche in Kadenz ................... 69, 70
Zeilenumbrüche in Musik ohne Metrum .... 70
Zeilenumbrüche in nicht metrischer Musik .. 69
Zeilenumbruch, Ballen ...................... 78
Zeit (in der Partitur) ......................... 111
Zentrieren von Text auf der Seite .......... 230
zentrierte Dynamik für Klaviermusik .... 304
Ziernoten ..................................... 105
<table>
<thead>
<tr>
<th>Thema</th>
<th>Seite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zitieren von anderen Stimmen</td>
<td>195, 198</td>
</tr>
<tr>
<td>zitiertes Text</td>
<td>216</td>
</tr>
<tr>
<td>Zurücksetzen von</td>
<td>63</td>
</tr>
<tr>
<td>Taktart-Standardeigenschaften</td>
<td>63</td>
</tr>
<tr>
<td>Zusammenfalten von Pausen</td>
<td>60</td>
</tr>
<tr>
<td>Zusammengesetzte Taktarten</td>
<td>73</td>
</tr>
<tr>
<td>Zusammenstoße</td>
<td>161</td>
</tr>
<tr>
<td>Zusammenstoße zwischen Systemen</td>
<td>305</td>
</tr>
<tr>
<td>Zusammenstoße, ignoriert</td>
<td>156</td>
</tr>
<tr>
<td>Zusammenstoße, kollidierende Notenkolumnen</td>
<td>156</td>
</tr>
<tr>
<td>Zusammenstoße, Taktnummern</td>
<td>101</td>
</tr>
<tr>
<td>Zusammenstoße, vertikal, vermeiden</td>
<td>535</td>
</tr>
<tr>
<td>Zwölftonmusik, Versatzzeichensstil</td>
<td>29</td>
</tr>
<tr>
<td>zweite Klammer</td>
<td>136</td>
</tr>
<tr>
<td>Zwischensystem-Tremolo</td>
<td>151</td>
</tr>
<tr>
<td>Zwischensystem-Klammer-Arpeggio</td>
<td>133</td>
</tr>
</tbody>
</table>