

Guido Gonzato

## Musik setzen mit ABC PLUS



Einführung in die ABCPLUS Notation und Applikation

Originaltitel: *Making Music with ABC PLUS*

Copyright Guido Gonzato, 2003

Übersetzt ins Deutsche von Frank-Michael Krause, E-Mail Mich.Krause@web.de

April 2004

Dieses Dokument wird unter den Bedingungen der GNU Public License (GPL) veröffentlicht.

Der Inhalt dieses Dokumentes wurde mit großer Sorgfalt erarbeitet. Der Autor und der Übersetzer können jedoch keinerlei Haftung für fehlerhafte Angaben und deren Folgen übernehmen.

Die in diesem Dokument erwähnten Soft- und Hardwarebezeichnungen sind in vielen Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen.

Warenzeichen:

Microsoft, MS-DOS, Windows sind eingetragene Warenzeichen der Microsoft Corporation.

Adobe, Postscript, Acrobat Reader sind eingetragene Warenzeichen von Adobe Systems Inc.

1	Teil I Computermusik mit ABCPLUS .....	6
1.1	Software.....	6
1.2	Motivation.....	8
1.3	Anforderungen .....	9
1.4	Kurzübersicht.....	9
1.5	Installation der Programme.....	9
1.6	ABC PLUS in a Nutshell.....	10
1.7	Ein erstes Notenblatt .....	10
1.8	Benutzung der Programme von der Kommandozeile .....	12
2	Teil II Melodie.....	14
2.1	Tonhöhe .....	14
2.2	Notenlänge: L: .....	15
2.3	Pausen und Abstände: z Z x y.....	16
2.4	Versetzungszeichen: ^ _ =.....	16
2.5	Punktierte Noten - gebrochener Rhythmus: < > .....	17
2.6	Haltebögen, Bindebögen und Staccato: - () .....	17
2.7	Tuplets (Triolen etc.) : (n.....	18
2.8	Akkorde: [ ].....	18
2.9	Liedtexte: w: W:.....	19
2.10	Sonderzeichen .....	20
2.11	Verzierungen.....	21
2.12	Ausdrucksymbole: !symbol! .....	21
2.13	Nutzerdefinierte Symbole.....	22
2.14	Zeilenumbrüche erzwingen: !.....	23
2.15	Zeilenumbrüche verhindern: \.....	24
3	Gestaltung und Strukturierung.....	25
3.1	Notenschlüssel und Tonarten.....	25
3.2	Taktart: M: .....	27
3.3	Taktstriche und Wiederholungen .....	27
3.4	Titel, Komponist, Tempo: T:, C:, Q:.....	28
3.5	Parts: P:.....	29
3.6	Begleitakkorde .....	30
3.7	Textergänzungen: “^_<>@” .....	31
3.8	Informationsfelder.....	31
4	Teil III Harmonie - Polyphonie in ABC PLUS .....	33
4.1	Stimmen und Notensysteme .....	33
4.2	Notenzeilen positionieren: %%staves .....	35
4.3	Stimmteilung.....	39
4.4	Wechsel des Notensystems.....	40
5	Teil IV Seitenlayout.....	43
5.1	Formatierungsparameter.....	43
5.2	Änderung von Parametern .....	45
5.3	Verwendung von Schriftarten .....	46
5.4	Unterbrechung des Notensystems .....	48
5.5	Mehrspaltiger Notensatz.....	48
5.6	Kopf- und Fußzeilen.....	50
5.7	Einfügen von Grafiken .....	51
5.8	Format Dateien.....	51
5.9	Numerierung von Takten und Seiten .....	52
5.10	Platz sparen.....	52

5.11	Anpassungen durch fortgeschrittene Nutzer.....	53
5.11.1	Neue Postscript Routinen.....	53
5.11.2	Begleitakkorde in italienischer Notation .....	53
5.11.3	Neue Symbole .....	54
5.11.4	Hinzufügen neuer Schriftarten .....	55
5.12	Flöten Fingersätze (tin whistle) .....	57
6	Teil V abcMIDI .....	58
6.1	Erzeugen von MIDI-Daten .....	58
6.1.1	Stimmen und Instrumente .....	58
6.1.2	Der Bass Schlüssel .....	59
6.1.3	Begleitakkorde .....	60
6.1.4	Neue Begleitakkorde .....	63
6.1.5	Schlagzeug .....	63
6.1.6	Schlaginstrumente .....	64
6.2	Erweiterte Nutzung von P: .....	65
6.2.1	Wiederholungen .....	66
6.3	midi2abc .....	67
6.4	yaps.....	68
6.5	Unterschiede und Inkompatibilitäten .....	68
7	Teil VI Automatische Quelltext Umwandlung.....	70
7.1	Der abcpp Präprozessor.....	70
7.1.1	Grundsätzliche Benutzung .....	70
7.1.2	Weitergehende Möglichkeiten .....	71
7.2	abc2abc .....	73
8	Teil VII Weitere Möglichkeiten .....	75
8.1	Nutzung von Notenblättern in anderen Programmen.....	75
8.1.1	Einfügen von Musik in L <sup>A</sup> T <sub>E</sub> X.....	75
8.2	Umwandlung von Grafiken ins EPS-Format .....	76
8.3	Stimmen extrahieren .....	76
8.4	Grenzen von abcm2ps .....	76
8.5	Abschließende Bemerkungen .....	77
8.5.1	Bitte eine kleine Spende .....	77
8.5.2	Im Gedenken an Annarosa Del Piero, 1930–2000.....	77
Teil VIII	Anhang .....	78
Anhang A	ABC PLUS Felder .....	78
Anhang B	Glossar .....	78
Anhang C	Zeichensätze .....	79
Anhang D	Formatierungsbefehle .....	79
D1	Seitenformat .....	79
D2	Text.....	80
D3	Schriftarten.....	80
D4	Abstände .....	81
D5	Andere Befehle.....	82
D6	abcMIDI Befehle.....	83
Anhang E	Postscript Schriftarten.....	84
Anhang F	MIDI Instrumente .....	84
Anhang G	Schlaginstrumente .....	86

*In loving memory of Annarosa, who introduced me to music;  
to my son Lorenzo, who will become a better musician than I am.*

# 1 Teil I Computermusik mit ABCPLUS

Falls Sie ein Musiker sind und wissen, wie man einen Computer nutzt, dürfen Sie sich freuen. Erstens, weil Sie ein Musiker sind und zweitens, weil ein Computer ein elegantes Werkzeug zum schreiben von Musik ist. Es gibt viele Programme, mit den Noten gesetzt werden können.

Die meisten dieser Programme verwenden eine graphische Oberfläche: Ein Notensystem wird angezeigt und der Nutzer editiert die Noten mit der Maus.

Ein alternativer Ansatz, der auch dieser Anleitung zugrunde liegt, ist das (be-)schreiben von Musik mittels Text. Noten und andere Symbole werden durch Buchstaben dargestellt. Ein Programm übersetzt diese Beschreibung in ein graphisches Ausgabeformat mit traditioneller Notation, wie z.B. Postscript oder PDF. Zu Gehör kann die Musik durch die Umwandlung ins MIDI-Format und abspielen auf gängigen Player-Programmen gebracht werden.

→ Graphische Programme sind für Anfänger leichter und intuitiver zu bedienen, die textuelle Notation erlaubt jedoch eine schnellere Eingabe und hat weitere Vorteile.

Es wurden viele textuelle Notationsformen erfunden. ABC ist eine der besten: einfach zu erlernen und sehr leistungsfähig. Sie ist sehr beliebt in Folk-Music-Kreisen. Tausende von Liedern, geschrieben mit ABC, sind im Internet verfügbar.

Der Zweck dieser Anleitung ist es, den Leser in die ABC PLUS Notation (Eine erweiterte Variante von ABC, mit welcher polyphone Musik gesetzt werden kann) und die zugehörigen Programme einzuführen.

Wenn Sie Noten für Ihren Chor oder Ihre Band setzen und drucken möchten, oder ein Liederbuch erstellen möchten oder Musiklehrer sind, dann haben sie das richtige Werkzeug gefunden.

## 1.1 Software

Die offizielle Webseite ist <http://abcplus.sourceforge.net>. Dort können Programme und Dokumentation heruntergeladen werden. Die Original-Sebseite für ABC-Notation ist <http://www.gre.ac.uk/~c.walshaw/abc>. Dort sind Links zu Liedsammlungen, Software und Dokumentation zu finden.

Eins der besten Programme zur Konvertierung von ABC PLUS Musik in Notenblätter ist abcm2ps, eine freies Programm unter der GNU GPL Lizenz. Das kleine Programm (nur 200 kByte) erzeugt hervorragende Notenblätter und bietet umfangreiche Möglichkeiten zur Beeinflussung des Layouts.

abcm2ps liest ABC PLUS-Dateien ein und konvertiert diese ins Postscript-Format, welches mit einem anderen freien Programm auf dem Bildschirm dargestellt und mit dem Drucker zu Papier gebracht werden kann: Ghostscript. Diese Applikation ist in der Lage POSTSCRIPT-Dateien in verschiedenste Formate umzuwandeln, u.a. auch in das PDF-Format für Adobes Acrobat Reader.

Der Autor von abcm2ps, Jean-François Moine, gibt "stabile" ('stable') und Entwicklungsversionen ('development') seines Programmes frei. Entwicklungsversionen werden fast wöchentlich freigegeben und haben jeweils viele Verbesserungen. Die letzte (zur Zeit der Erstellung dieser Anleitung) als stabil freigegebene Version ist 3.7.17, freigegeben im Oktober 2003.

Ein weiteres sehr nützliches Programm ist abc2midi, welches ABC PLUS-Dateien in das MIDI-Format konvertiert. Die Benutzung dieses Programmes wird in Teil V dieses Anleitungs beschrieben.

*Hinweis: Es gibt viele weitere Programme, welche die Speicherung von Musik im ABC-Format erlauben, u.a. einige mit graphischer Oberfläche. Einige davon sind sehr interessant und erlauben die Erstellung von Notenblättern ohne das die Erstellung von ABC PLUS-Dateien mit Hilfe dieser Anleitung erlernt werden muss... Diese Programme sollen hier jedoch nicht weiter beschrieben werden. Lediglich ein Programm soll kurz Erwähnung finden, weil es wirklich gut ist: Note Editor, <http://tan.informatik.tu-chemnitz.de/~jan/noteedit/noteedit.html>.*

Alle Programme sind Kommandozeilen-Programme, sie können nicht gestartet werden, indem auf ein Desktop-Symbol doppelgeklicked wird.

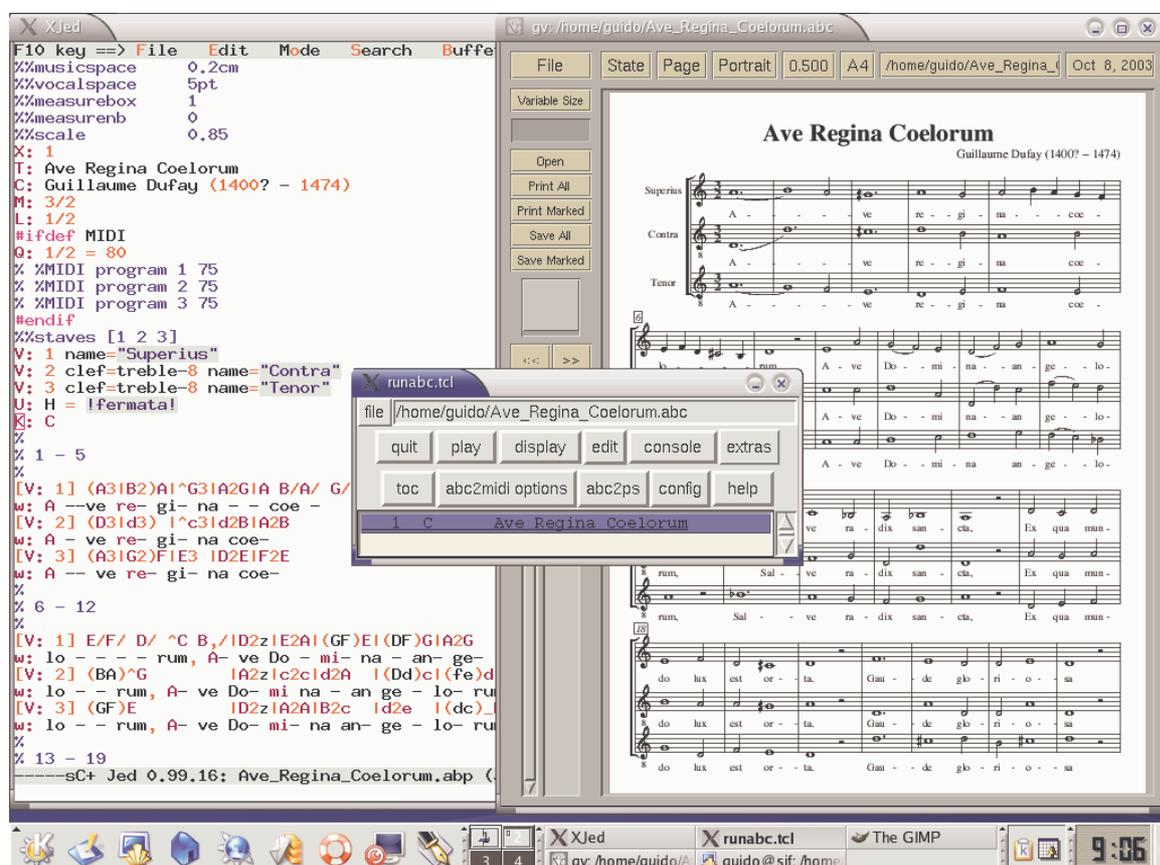
Es muss ein Kommandozeilen Fenster (DOS-Box oder Linux Terminal) geöffnet werden, in welchem Befehle eingegeben werden – richtig wie in alten Tagen. Die Benutzung der Kommandozeile kann jedoch auch umgangen werden.

Um ABC PLUS-Dateien zu erstellen, wird ein Texteditor benötigt. Man kann mit einfachen Programmen wie Windows Notepad arbeiten, jedoch vereinfacht ein angepasster Editor wie raunabc.tcl oder JEDABC die Arbeit erheblich.

Hier sei die Verwendung von JEDABC empfohlen.

JEDABC kennzeichnet die Noten durch Verwendung verschiedener Farben, spielt die Noten beim eingeben, ruft automatisch den POSTSCRIPT-Converter auf und vieles mehr. Die beste Funktion von JEDABC ist vielleicht, daß die Benutzung der Kommandozeile nicht nötig ist. Dazu im Detail später.

runabc.tcl ist als Editor weniger leistungsfähig, schlägt JEDABC jedoch eindeutig was die Interaktion mit den einzelnen ABC PLUS Applikationen anbetrifft. JEDABC und runabc.tcl können sich jedoch gut gegenseitig ergänzen.



## 1.2 Motivation

Aus Erfahrung weiß ich, dass Programme mit graphischer Oberfläche (fast alle kommerziellen Programme) als einfacher zu benutzen gelten als Programme ohne graphische Oberfläche. Warum sollte man also ABC PLUS lernen wenn der Notensatz mit einem graphischen Programm einfacher erscheint?

Verglichen mit einem graphischen Programm hat ABC PLUS jedoch viele Vorteile:

- **Leistungsfähigkeit:** Mit ABC PLUS können Noten von sehr einfachen Liedern bis hin zu komplexen vielstimmigen Sätzen erstellt werden.
- **Qualität:** Die Qualität der Notenblätter, die mit ABC PLUS erzeugt werden können, ist exzellent.
- **Preis:** Während kommerzielle Programme oft sehr teuer sind, sind die meisten Programme, die im Zusammenhang mit ABC PLUS eingesetzt werden können, freie Software und können frei weitergegeben werden.
- **Geringe Anforderungen:** ABC PLUS Programme sind sehr kompakt und können auch auf älteren Computern ausgeführt werden.
- **Portabilität:** Die Musik wird als Text-, POSTSCRIPT-, PDF- oder MIDI-Datei erstellt und nicht in einem proprietären Dateiformat. Damit kann die Musik einfach weitergegeben werden und nicht nur von Leuten verarbeitet werden, die die richtige Software haben, um damit umzugehen.
- **Einfachheit:** ABC PLUS ist gut lesbar und nach ein bisschen Praxis kann es intuitive verwendet werden. Das Einbeziehen von Notenblättern in Webseiten oder in Textverarbeitungen ist einfach.
- **Flexibilität:** In vielen Fällen ist ABC PLUS flexibler als graphische Programme und einfacher zu benutzen.
- **Geschwindigkeit:** Das Schreiben von Musik mit ABC PLUS geht sehr schnell.
- **Lerneffekt:** Falls Sie ein Musiklehrer sind, ist ABC PLUS ein wertvolles Werkzeug, welches das Verständnis der Musiktheorie fördert.
- **Spaß:** Meiner Meinung nach macht das Schreiben von Musik mit ABC PLUS mehr Spaß als mit anderen Programmen.

Die Nachteile sollen jedoch auch nicht verschwiegen werden:

- **Lernkurve:** Während ein graphisches Programm es ermöglicht, sofort Ergebnisse zu erreichen (zumindest in der Theorie), erfordert ABC PLUS zuerst das Lesen der Dokumentation bevor mit dem Setzen von Noten begonnen werden kann.
- **Unterstützung fremder Dateiformate:** falls die Arbeitsumgebung es erfordert, mit speziellen kommerziellen Programmen zu arbeiten, könnte es schwierig sein, vorhandene Musik-Dateien ins ABC PLUS-Format und umgekehrt umzuwandeln.
- **Begrenzungen:** ABC PLUS ist zur Zeit nicht in der Lage, einige besondere Arten von Musik darzustellen. Beispiel sind gregorianische Gesänge, Schlagzeug und Musik außerhalb Europas.

Um den ersten Punkt zu überwinden ist diese Anleitung hoffentlich ein guter Anfang. Es sei jedoch empfohlen, ebenfalls einige Beispiele fertiger Musik anzusehen, die mit ABC/ ABC PLUS erstellt wurde. Die ABC-Homepage hat viele Links zu ABC-Sammlungen.

Was ABC PLUS für den einzelnen bringt, mag jeder selbst nach Studium dieser Anleitung entscheiden. Ich gehe jedenfalls davon aus, dass Sie es nicht bereuen werden, die Zeit investiert zu haben !

## 1.3 Anforderungen

Benötigt wird ein PC mit Windows, Linux oder anderen Unix-Varianten. Der allgemeine Umgang mit einem Computer sollte ebenfalls vertraut sein. Kenntnisse der DOS- oder Linux-Kommandozeile sind jedoch nicht vonnöten.

Kenntnisse über die Grundregeln der Musiknotation sollten vorhanden sein.

(Anm. d. Übersetzers: Für die Einarbeitung in diese Thematik und Musiktheorie im Allgemeinen sei auf folgendes deutschsprachiges Buch verwiesen: Wieland Ziegenrucker: ABC Musik)

*Warnung: Ich bin (bis jetzt) kein professioneller Musiker und meine musikalische Ausbildung kommt aus der klassischen Musik. Die ABC Notation (von der ABC PLUS abgeleitet ist) ist vorwiegend bei angelsächsischen Folk Musikern populär. Ich weiß nicht sehr viel über diese Musik. Deshalb spiegelt diese Anleitung vorwiegend meine Vision darüber wieder, was mit ABC PLUS möglich ist.*

## 1.4 Kurzübersicht

Das Schreiben von Musik mit ABC PLUS gliedert sich in folgende Schritte:

1. Mit JEDABC oder einem anderen Editor den ABC PLUS-Quelltext für das Musikstück schreiben.
2. Mit abcm2ps aus dem Quelltext das Notenblatt im POSTSCRIPT Format erzeugen.
3. Mit Ghostscript die POSTSCRIPT-Datei ansehen.
4. Aus der POSTSCRIPT-Datei mit Ghostscript/Ghostview eine PDF-Datei erzeugen.
5. Wenn gewünscht mit abc2midi eine MIDI-Datei erzeugen.
6. Wenn die geschriebene Musik nicht urheberrechtlich geschützt ist, das ABC File im Internet zur Verfügung stellen, damit sich auch andere daran erfreuen können.

## 1.5 Installation der Programme

Im folgenden sind die Hompages der Programme angegeben:

- das abcm2ps Notensatzprogramm:  
<http://moinejf.free.fr>, <http://abcplus.sourceforge.net>
- GhostScript and GhostView:  
<http://www.cs.wisc.edu/~ghost>  
Es sind die Dateien gs811w32.exe und gsv45w32.exe zu installieren (die jeweils aktuellen Versionsnummen können abweichen)
- der Jed editor:  
<http://www.jedsoft.org/jed>
- JEDABC:  
<http://abcplus.sourceforge.net/#JedABC>
- abc2midi:  
<http://ifdo.pugmarks.com/~seymour/runabc/top.html>,  
<http://abcplus.sourceforge.net>
- runabc:  
<http://ifdo.pugmarks.com/~seymour/runabc/runabc.html>

Nach dem Download der Programme sind zunächst GhostScript und GhostView zu installieren.

Um JEDABC zu installieren, ist den Anweisungen zu folgen, die dem Programm beiliegen. Die Programme abcm2ps und abc2midi haben keine Insatalltionsroutine. Die Dateien sollten in den Ordner c:\windows\command (Windows 95/98/ME) oder c:\winnt\system32 (Windows NT/2000/XP) kopiert werden oder alternativ in ein anderes Verzeichnis, das in der Systemvariable PATH enthalten ist.

Für die Installation unter Windows NT/2000/XP sind Administratorrechte nötig!

*Hinweis: Es ist empfohlen, an dieser Stelle zunächst die Dokumentation für JEDABC zu studieren bevor diese Anleitung weiter durchgearbeitet wird!*

*Insbesondere sollte die Autobar-Funktion ausgeschaltet werden, wenn die ersten ABC PLUS-Dateien geschrieben werden.*

## 1.6 ABC PLUS in a Nutshell

ABC PLUS Musik wird in Textfiles mit der Erweiterung .abc oder .abp gespeichert. Das ist nicht zwingend erforderlich, jedoch empfohlen.

ABCPLUS benutzt die Zeichen, die auf jedem Computerkeyboard zu finden sind, um Noten und Symbole zu repräsentieren:

- Buchstaben wie A B C D a b c d z repräsentieren Noten und Pausen.
- Bindebögen, Vorzeichen werden mit = - \_ ( ) repräsentiert
- Ausdruckssymbole werden mit Sequenzen wie !ff!, !fermata!, !tenuto! notiert
- Taktart, Titel, Tonart werden mit Feldern beschrieben - M:, T:, ...
- Formatierungsdetails und -Optionen sowie MIDI-Optionen werden mit Commands wie %%titlefont oder %%MIDI program 19 beschrieben.

Alle musikalischen Merkmale werden durch Zeichen-Sequenzen beschrieben.

Ein Lied besteht im Sinne von ABC aus zwei Teilen, einem Kopf und dem Rumpf.

Der Kopf (engl. Header) enthält Informationen über das Lied wie z.B. Titel, Autor, Tonart, Taktart usw. Diese Informationen werden in Feldern(engl. Fields) beschrieben. Der Rumpf(engl. Body) enthält die Noten, Symbole und Liedtext.

## 1.7 Ein erstes Notenblatt

Als erstes Beispiel wollen wir die C-Dur-Tonleiter schreiben und uns den Quelltext ansehen. Erzeuge eine Datei scale1.abc mit einem Texteditor und kopiere den folgenden Text hinein:

```
X: 1 % start of header
K: C % scale: C major
% C major, two octaves
C D E F G A B c | c d e f g a b c' |
% end of file scale1.abc
```

Auf der Kommandozeile führe das Übersetzungsprogramm abcm2ps aus um ein Postscriptfile zu erzeugen:

```
abcm2ps -O= -c scale1.abc
```

Mittels GhostView oder einem anderen Postscript-Viewer kann die die erzeugte Datei angezeigt und in ein PDF-File umgewandelt werden.

Die Anzeige in GhostView sollte wie folgt aussehen:



Erzeugung eines PDF-Files mit GhostView:

- Windows Nutzer wählen `File/Convert...` in GhostView, setzen `pdfwrite` als output type und wählen eine dpi-Auflösung. Empfohlen wird 600 dpi.
- Unter Linux und anderen Unix-Varianten kann folgender Befehl zur Erzeugung des PDF-Files verwendet werden: `ps2pdf scale1.ps`

Jetzt fügen wir einen Fehler in den Quelltext ein und wollen sehen, was passiert: Wir tauschen den ersten Taktstrich | gegen das Zeichen #. Nach Speicherung und dem Versuch, den Quelltext zu übersetzen erhalten wir folgende Fehleranzeige:

```
Error in line 4:16: Bad character
```

Nach Korrektur und Neuübersetzung ist alles wieder in Ordnung.

Nun zum Aufbau des Quelltextes. Zu Beginn finden wir zwei Felder im Header: X:, index , und K:, Tonart , alles in Großbuchstaben. Dies sind die unbedingt erforderlichen Felder. Hinter X: wird immer eine Zahl angegeben, welche das Musikstück innerhalb der Quelldatei kennzeichnet (Es können mehrere Stücke innerhalb einer Datei geschrieben werden). Das %-Zeichen beginnt einen Kommentar. Der Rest der Zeile wird vom Übersetzungsprogramm ignoriert.

Die Felder können Leerzeichen enthalten. X: 1 und X: 1 sind gleichwertig.

Das K: Feld legt die Tonart für das Stück fest. C steht für C-Dur.

Das X:-Feld muss das erste Feld im Header sein, das K:-Feld das letzte. Andere Felder können in beliebiger Folge zwischen X: und K: angeordnet werden.

Die nächste Zeile enthält die Noten. Großbuchstaben werden für die Noten der Zentralen Oktave verwendet, Kleinbuchstaben für die nächsthöhere Oktave. Das Zeichen 1' wird verwendet, um einen Taktstrich einzufügen. Taktstriche können an beliebigen Stellen eingefügt werden !

Das Apostroph beim höchsten c' kennzeichnet Noten der nächsthöheren Oktave.

Das Übersetzungsprogramm `abcm2ps` setzt die Taktart automatisch auf 4/4 und die Standard-Notenlänge auf Achtelnote, wenn nichts anderes angegeben ist.

Das war nicht weiter schwierig bis jetzt. Wir sind nun bereit, uns mit weiteren Details des Notensatzes vertraut zu machen, nach dem wir nun einige Übungen mit dem bisherigen Wissen durchführen.

Versuchen Sie, einige ABCPlus-Files zu schreiben und zu Übersetzen. Benutzen Sie zufällige Notensequenzen, aber üben Sie das schreiben, speichern, übersetzen und Anzeigen der Noten.

## 1.8 Benutzung der Programme von der Kommandozeile

Wollen Sie einen anderen Editor benutzen als JED, der bereits die Kommandos zur Übersetzung und Notenanzeige in das Menu integriert hat, so haben Sie die Möglichkeit, das Übersetzungsprogramm direkt von der Kommandozeile aus aufzurufen. Um unter Windows 95 oder 98 auf die Kommandozeile zu kommen geben Sie im Startmenu unter Ausführen . . . den Befehl `Command` ein. Windows NT, 2000 oder XP Nutzer geben hier `cmd` ein, um ein Kommandozeile Fenster zu öffnen.

Nun muss mittels `cd` (`change directory`) in das Verzeichnis gewechselt werden, in dem der Quelltext abgespeichert wurde, z.b.

```
cd \daten\music\abc ↵
```

Dann sollte sichergestellt sein, daß der Speicherort des Übersetzungsprogrammes `abcm2ps` in der `PATH`-Variable des Systems enthalten ist. Überprüfen Sie dies, indem sie auf der Kommandozeile

```
PATH ↵
```

eingeben. Die Ausgabe sollte den Speicherort von `abcm2ps` enthalten.

Der Inhalt der Variablen `PATH` kann bei Windows XP über die Systemsteuerung unter System/Erweitert -> Button Umgebungsvariablen geändert werden.

Alternativ kann das Programm `abcm2ps` mit in das Verzeichnis der Quelltexte kopiert werden. Dann entfällt die Erweiterung der `PATH`-Variable.

Haben Sie den Quelltext fertig für die Übersetzung, so geben sie folgendes Kommando ein:

```
c:\daten\music\abc>abcm2ps scale1.abc ↵
```

Sie sollten folgende Ausgabe erhalten:

```
abcm2ps-3.7.1 (August 3, 2003)
File scale1.abc
Output written on Out.ps (1 page, 1 title, 16555 bytes)
c:\daten\music\abc>
```

Die Datei `Out.ps` enthält die Noten im Postscript-Format.

Enthält die Datei Fehler, so werden in der Ausgabe Fehler angezeigt:

```
c:\daten\music\abc> abcm2ps scale1.abc
abcm2ps-3.7.1 (August 3, 2003)
File scale1-it.abc
Error in line 4.16: Bad character
4 C D E F G A B c # c d e f g a b c' |
      ^
Output written on Out.ps (1 page, 1 title, 16555 bytes)
c:\daten\music\abc>
```

Die Stelle im Quelltext, an der der Fehler auftrat, wird mit einem Pfeil ^ angezeigt. In diesem Fall wurde der Fehler ignoriert und trotzdem eine Ausgabedatei erzeugt. Bei schwereren Fehlern kann unter Umständen keine Ausgabedatei erzeugt werden.

## 2 Teil II Melodie

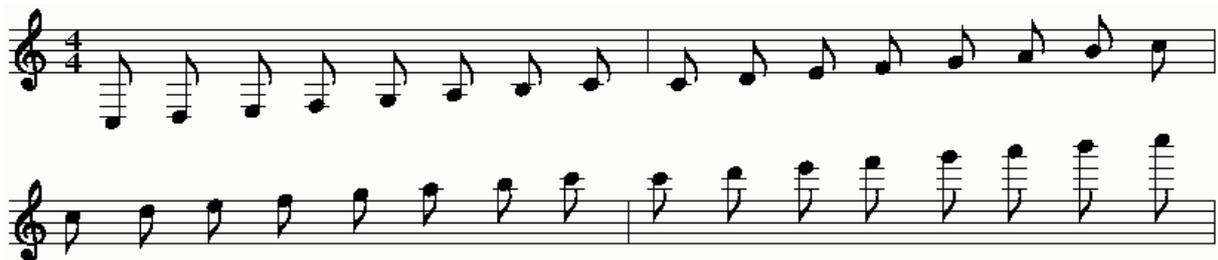
Dieser Teil der Anleitung beschäftigt sich mit den wesentlichen Eigenschaften von Noten: Tonhöhe, Länge, Betonung, Punkte, Bögen, Vorschläge und Ausdruckssymbole.

### 2.1 Tonhöhe

Der folgende Quelltext (`scale2.abc`) zeigt wie man Noten über und unter dem Notensystem darstellt. Es handelt sich wiederum um die C-Dur Tonleiter. Anstatt nur `K:C` zu schreiben wie im vorangegangenen Beispiel notieren wir nun `K:C treble`. Neben der Definition der Tonart auf C-Dur wird damit der Violin-Schlüssel für das Notensystem festgelegt.

In diesem Beispiel muss dies so festgelegt werden, da sich viele Noten weit unterhalb des Notensystems befinden, wodurch `abcm2ps` automatisch den Bass-Schlüssel benutzen würde. Der Quelltext muss ohne die Option `-c` übersetzt werden.

```
X: 1
K: C treble
% C major, four octaves:
C, D, E, F, G, A, B, C | C D E F G A B c |
c d e f g a b c' | c' d' e' f' g' a' b' c'' |
```



Die Notation folgt der Regel: Werden nach einem Notenzeichen ein oder mehrere Kommas geschrieben, so bedeutet das eine Versetzung der Note um eine oder mehrere Oktaven nach unten. Werden statt des Komma ein oder mehrere Apostroph geschrieben, so bedeutet dies eine Verschiebung um eine oder mehrere Oktaven nach oben.

Im Quelltext wurde die Tonleiter in zwei Zeilen notiert. Die Ausgabe erfolgt ebenfalls in zwei Notenzeilen. Dies ist eine der Grundregeln der ABCPlus-Notation: eine neue Zeile im Quelltext bedeutet ebenfalls eine neue Zeile in der Ausgabe. Ausnahmen zu dieser Regel werden später behandelt.

In obigem Beispiel ist der Notenabstand zu groß. Alle Noten in einer Zeile würde wahrscheinlich besser aussehen. Mit der Option `-c` kann `abcm2ps` angewiesen werden, Zeilenumbrüche zu ignorieren und die Takte optimal zu formatieren:

Die Ausgabe von `abcm2ps -O= -c scale2.abc` sieht wie folgt aus:



Ein Notenblatt mit Bass-Schlüssel lässt sich mit folgendem Quelltext erzeugen:

```
X: 1
L: 1/4
K: C
C,, , D,, , E,, , F,, , | G,, , A,, , B,, , C, | C, D, E, F, | G, A, B, C |
```



**Achtung:** Da das Programm aus dem englischsprachigen Raum kommt, erfolgt die Eingabe der Noten nach dem englischsprachigen System. Die im deutschen Sprachraum mit 'h' bezeichnete Note wird als 'b' eingegeben und die im deutschen als 'b' bezeichnete Note wird als mit Vorzeichen b versehenes 'b' eingegeben, also in der ABC PLUS Syntax als '\_b' (siehe auch Kap. 2.4).

## 2.2 Notenlänge: L:

Wenn nicht anders angegeben, wird die Standard-Notenlänge in Abhängigkeit von der Taktart gesetzt: Ist das Metrum größer oder gleich  $\frac{3}{4}$ , so ist die Standard-Notenlänge  $\frac{1}{8}$ , ist es kleiner, z.B.  $\frac{2}{4}$ , so ist die Standard-Notenlänge  $\frac{1}{16}$ .

Das L:-Feld wird benutzt, um die Standard-Notenlänge zu definieren. Im obigen Beispiel wird sie auf Viertelnote gesetzt.

Um die Länge einer Note zu verlängern wird ein Faktor hinter das Notenzeichen geschrieben. Um kürzere Noten zu schreiben wird ein Divisor hinter das Notenzeichen gesetzt:  $/2$ ,  $/4$ ,  $/8$  oder alternativ  $/$ ,  $//$ ,  $///$ . Leerzeichen zwischen Notenzeichen und Multiplikator oder Divisor sind nicht erlaubt!

Beispiel:

```
X: 1
L: 1/4
K: C
C16|C8|C4|D2 D2|E0 E E E E|FF F/ F/ F/ F/|
G3 G// G/4 G/8 G/8 G/16 G/16 G/16 G/16|
```



Abcm2ps erlaubt die Definition von Noten länger als eine ganze Note. Die erste Note im Beispiel wird *longa* genannt und entspricht der Länge von 4 ganzen Noten. Die zweite Note wird *brevis* genannt und entspricht der Länge von 2 ganzen Noten.

Der Abstand zwischen zwei Noten ist proportional zu ihrer Länge.

Eine weitere besondere Note ist das erste E in Takt 5. Der Längenmultiplikator 0 führt zu einem ausgefüllten Notenkopf ohne Hals.

Leerzeichen zwischen Noten und Taktstrichen können im Quelltext beliebig zur Verbesserung der Lesbarkeit des Quelltextes eingefügt werden, sofern die Notenlänge größer einer Achtel

Note ist. Ist die Notenlänge eine Achtel oder kürzer, so können Leerzeichen zwischen den Noten nicht beliebig gesetzt werden: Werden keine Leerzeichen zwischen die Noten gesetzt, so werden sie mit einem Balken verbunden. Siehe dazu auch folgendes Beispiel:

```
X: 1
K: C
C D E F CDEF | C D E F C/D/E/F/G/A/B/c/ |
c/B/A/G/ F/E/D/C/ C4 |
```



### 2.3 Pausen und Abstände: z Z x y

Für Pausen wird der Buchstabe z verwendet. Für die Pausenlänge gelten die gleichen Regeln wie für Notenlängen.

Um Pausen über mehrere Takte zu notieren, wird der Buchstabe Z, gefolgt von der Anzahl der Takte, verwendet.

```
X: 1
L: 1/4
K: C
Z12|z16|z8|z4|C2 z2|C z C z|C z/ z/ C z// z// z// z//|
```



Mit den Buchstaben x und y lassen sich unsichtbare Pausen (x) und zusätzliche Abstände (y) realisieren:

```
X: 1
L: 1/4
K: C
C D E/E/E/E/ F/F/F/F/|C D E/yyE/yyE/yyE/ F/yF/yF/yF/ yyy|xxxG|
```



Unsichtbare Pausen werden oft in der Notation von Klaviermusik verwendet. Beispiele dafür werden in Kap. 4.2 gegeben.

### 2.4 Versetzungszeichen: ^ \_ =

Ein Kreuz wird als ^ vor der Note notiert, ein b als \_ und ein Aufhebungszeichen als =. Leerzeichen zwischen Versetzungszeichen und Note sind nicht erlaubt.

Als Beispiel sei die chromatische Tonleiter dargestellt:

```
X: 1
L: 1/4
K: C
C ^C D ^D | E F ^F G | ^G A ^A B | c^c=cz |
c B _B A | _A G _G F | E _E D _D | C_C=Cz |
```



```
X: 1
L: 1/4
K: C
.C/ .C/ D - D .E/ .E/|EF-FG| (C/E/G/c/) (c/G/E/C/) |C2 z2||
```



## 2.7 Tuplets (Triolen etc.) : (n

Triolen, Quartolen, Quintolen usw. werden erzeugt, indem nach einer öffnenden runden Klammer unmittelbar die Zahl der Noten des Tuplets und danach die Noten angegeben wird. Leerzeichen sind auch hier nicht erlaubt.

Es gibt folgende Varianten:

- (2 zwei Noten verteilt auf die Dauer von drei Noten
- (3 drei Noten verteilt auf die Dauer von zwei Noten
- (4 vier Noten verteilt auf die Dauer von drei Noten
- (5 fünf Noten verteilt auf die Dauer von n Noten → siehe unten
- (6 sechs Noten verteilt auf die Dauer von zwei Noten
- (7 sieben Noten verteilt auf die Dauer von n Noten
- (8 acht Noten verteilt auf die Dauer von drei Noten
- (9 neun Noten verteilt auf die Dauer von n Noten

Wenn die Taktart 6/8, 9/8, 12/8 usw. ist, so ist n gleich 3, ansonsten 2.

```
X: 1
M: 2/4
L: 1/8
K: C
(3cde e2 | (3cde (3def | (3cze c(3(d/e/f/)|
(6cegzeg g2| (6cegzcg (3ceg |]
```



## 2.8 Akkorde: [ ]

Akkorde werden geschrieben, indem die Akkordnoten in eckige Klammern eingfasst werden. Leerzeichen zwischen Klammern und Noten sind auch hier wieder nicht erlaubt.

Ein Akkord verhält sich bezüglich Bögen, Punkten etc. wie eine einzelne Note.

Um zwei Akkorde mit Haltebögen zu versehen, ist hinter jeder Akkordnote ein – zu schreiben.

```
X: 1
L: 1/4
K: C
CE [C2G] c | .[CEGc] [C2D2G2c2] ([C/E/G/c/] [E/a/B/e/]) |
D^FA| [D^FA] [D^FA] > [D^FA] [D^FA] | [C2-E2-G2-] [CEG] z |
```



**Hinweis:** Die hier beschriebene Akkord-Notation sollte nicht für etwas völlig anderes, wie im folgenden Beispiel dargestellt, missbraucht werden!



Dies sind keine Akkorde, sondern verschiedene Stimmen im gleichen Notensystem. Im Kap. 4.3 wird darauf näher eingegangen.

Besondere Beachtung verdient der Akkord im ersten Takt. Obwohl abcm2ps Akkorde mit unterschiedlich langen Noten unterstützt, ist diese Notation nicht empfohlen, da andere ABCPLUS Programme diese nicht unterstützen.

## 2.9 Liedtexte: w: W:

Liedtexte können am Ende eines Liedes hinzugefügt werden oder unterhalb jedes Notensystems den einzelnen Noten zugeordnet werden.

Im ersten Fall werden die Textzeilen am Ende des Liedes angefügt. Dabei muss jede Zeile mit W: beginnen.

Den einzelnen Noten zugeordneter Text ist komplexer. Unmittelbar nach jeder Notenzeile wird in diesem Fall eine oder mehrere Zeilen, beginnend mit w: , geschrieben. Nach w: folgt die Textzeile, aufgeteilt in Silben. Die Zuordnungsregeln sind wie folgt:

- Das Zeichen – separiert die einzelnen Silben eines Wortes. Wenn vor dem Bindestrich ein Leerzeichen eingefügt wird, so wird eine Note übersprungen. Ein Leerzeichen nach einem Bindestrich wird ignoriert.
- | springt zum nächsten Takt
- \_ wird verwendet, um die Silbe auf eine weitere Note auszudehnen, es wird eine horizontale Linie gezeichnet.
- \* überspringt eine Note
- ~ ordnet zwei Silben einer Note zu
- \- fügt ein Bindestrich in die Ausgabe ein
- \ Zeilenumbruchszeichen, die nächste w:-Zeile führt die mit \ beendete Zeile fort und gehört zur gleichen Notenzeile

Das folgende Lied ist „Happy Birthday“ auf italienisch:

```
X: 1
M: 3/4
K: C
C> C | D2C2F2 | E2-E z C> C | D2C2G2 | F2-F z C> C |
w: tan- ti~au- gu- ri a te, _ tan- ti~au- gu- ri a te, * tan- ti~au-
c2A2F2 | E2D z _B> B | A2F2G2 | F6 |]
w: gu- ri fe- li- ci, tan- ti~au- gu- ri a te!
%
W: Tanti auguri a te, tanti auguri a te,
W: tanti auguri felici, tanti auguri a te!
```

tan - ti au - gu - ri a te, — tan - ti au - gu - ri a te, tan - ti au - gu - - ri fe - - li - - ci, tan - ti au - gu - - ri a te!

Tanti auguri a te,  
tanti auguri a te,  
tanti auguri felici, tanti auguri a te!

Wenn eine w:-Zeile Ziffern enthält, so werden diese nicht den Noten zugeordnet, sondern ein wenig nach links gerückt. Damit können mehrere Strophen eines Liedes den Noten zugeordnet werden. Die Ziffer muss mit einer Tilde (~) mit der ersten Silbe verbunden werden.

Wenn Ziffern den Noten zugeordnet werden sollen, z.B. Fingersatz, so ist vor der Ziffer eine Tilde (~) zu schreiben.

```
X: 1
L: 1/4
K: C
CDEF|GABc|c2z2|z4|
w: 1.~do re mi fa sol la si do doooo
w: 2.~~~la la la la 1 2 ~3 ~4 laaaa
```

1. do re mi fa sol la si do doooo  
2. la la la la 1 2 3 4 laaaa

ACHTUNG: Es muss darauf geachtet werden, dass die Anzahl Silben der Anzahl der Noten entspricht!

## 2.10 Sonderzeichen

Solange Lieder in englischer Sprache geschrieben werden sollen, ist alles einfach. Schwierig wird es, wenn andere Sprachen benötigt werden. Dann müssen spezielle Zeichen dieser Sprachen gesetzt werden, die auf einem US-Keyboard nicht vorhanden sind.

Das Problem wird durch spezielle Zeichensequenzen gelöst. Beginnend mit einem \, dann ein spezielles Zeichen, dann der Buchstabe, der modifiziert werden soll. Das klingt zwar ziemlich kompliziert, ist es aber eigentlich nicht. In der folgenden Tabelle sind einige Beispiele für Sonderzeichen und die zugehörigen Zeichensequenzen dargestellt:

Letter	Sequence
à è á é	\ 'a \ 'e \ 'a \ 'e
Û ü ë ö	\ "U \ "u \ "e \ "o
Ñ ñ	\ ~N \ ~n
ß ø Ø å Å	\ ss \ /o \ /O \ aa \ AA
Ô ô ç Ç	\ ^O \ ^o \ cc \ cC
Æ Œ æ œ	\ AE \ OE \ ae \ oe

Eine komplette Tabelle mit allen unterstützten Symbolen ist in Anhang C enthalten.

## 2.11 Verzierungen

Die Tilde (~) beschreibt eine allgemeine Verzierung. Ihre Bedeutung und Ausführung hängt von der Interpretation des Musikers ab.

Um Vorschlagsnoten zu schreiben, werden eine oder mehrere Noten in geschweifte Klammern gesetzt. Um einen kurzen Vorschlag mit durchgestrichenem Hals zu erzeugen, ist hinter die öffnende geschweifte Klammer ein / zu schreiben, dann eine einzelne Note. Obwohl es wenig Sinn macht, kann die Notenlänge auch bei Vorschlagsnoten festgelegt werden.

```
X: 1
L: 1/4
K: C
~c2 {/d}c {c2d2}c|{d/c/d/}c {ede}d {fef}e f|
c/{gfef}d/e/f/ f/e/{gfedc}d/c/|c G E {cBAGFED}C|
```



Um die Bindebögen zwischen Vorschlagsnoten und der folgenden Note zu unterdrücken, kann `abcm2ps` mit der Option `-G` aufgerufen werden oder ein Formatierungsparameter im Quelltext gesetzt werden.

## 2.12 Ausdruckssymbole: !symbol!

Ausdruckssymbole werden in der allgemeinen Form `!symbol!` notiert. Sie werden unmittelbar vor der Note geschrieben, der sie zugeordnet werden sollen. Die Positionierung einiger Symbole ist abhängig davon, ob Textzeilen unter einer Notenzeile vorhanden sind oder nicht. Wenn Text vorhanden ist werden sie über der Notenzeile dargestellt, ansonsten darunter. In Kap. 5 ist dargestellt, wie die Positionierung gesteuert werden kann.

Folgendes Beispiel zeigt die meisten verfügbaren Symbole. Einige Symbole fehlen und werden in Kap. 5.11 näher beschrieben.

```
X: 1
T: Symbols
L: 1/4
K: C
!+!C!0!D!1!E!2!F |!3!C!4!D!5!E!D.C.!F|!D.S.!C!accent!D!breath!E!coda!F|
w: !+! !0! !1! !2! !3! !4! !5! !D.C.! !D.S.! !accent! !breath! !coda!
!crescendo(!C!crescendo)!D!diminuendo(!E!diminuendo)!F|
w: !crescendo(! !crescendo)! !diminuendo(! !diminuendo)!
!downbow!C!emphasis!D!fermata!E!f!F|!ff!C!fff!D!ffff!E!fine!G|
w: !downbow! !emphasis! !fermata! !f! !ff! !fff! !ffff! !fine!
!invertedfermata!C!longphrase!D!lowermordent!E
w: !invertedfermata! !longphrase! !lowermordent!
```

```

!mediumphrase!F|
!mf!C!mordent!D!open!E!p!F!|!pp!C!ppp!D!pppp!E!pralltriller!F|
w: !mediumphrase! !mf! !mordent! !open! !p! !pp! !ppp! !pppp!
!pralltriller!
!roll!E!segno!F!sfz!C!shortphrase!D!|!snap!E!tenuto!F!thumb!C!trill!D|
w: !roll! !segno! !sfz! !shortphrase! !snap! !tenuto! !thumb! !trill!
!turn!E!upbow!F!uppermordent!C!wedge!D|
w: !turn! !upbow! !uppermordent! !wedge!
!turnx!F!invertedturn!C!invertedturnx!D!arpeggio! [CEGc]!trill(!F|
!trill)!C2|]
w: !turnx! !invertedturn! !invertedturnx! !arpeggio! !trill(! !trill)!

```

## Symbols

The image shows six staves of musical notation in 4/4 time, illustrating various symbols and dynamics. Each staff has a corresponding list of symbols below it.

- Staff 1:** Shows notes with various articulation marks above them. Symbols: !+, !0, !1, !2, !3, !4, !5, !D.C., !D.S., !accent, !breath, !coda.
- Staff 2:** Shows notes with dynamic markings and articulation marks. Symbols: !crescendo(!, !crescendo), !diminuendo(!, !diminuendo), !downbow, !emphasis, !fermata, !f.
- Staff 3:** Shows notes with dynamic markings and articulation marks. Symbols: !ff, !fff, !ffff, !fine, !invertedfermata, !longphrase, !lowermordent, !mediumphrase!
- Staff 4:** Shows notes with dynamic markings and articulation marks. Symbols: !mf, !mordent, !open, !p, !pp, !ppp, !pppp, !pralltriller, !roll, !segno, !sfz, !shortphrase!
- Staff 5:** Shows notes with articulation marks. Symbols: !snap, !tenuto, !thumb, !trill, !turn, !upbow, !uppermordent, !wedge!
- Staff 6:** Shows notes with articulation marks. Symbols: !turnx, !invertedturn, !invertedturnx, !arpeggio, !trill(!, !trill)!

### 2.13 Nutzerdefinierte Symbole

Die meisten Ausdruckssymbole sind längere Wörter, welche den Quelltext schwer lesbar machen.

Um dieses Problem zu umgehen, kann ein einzelner Buchstabe für ein Symbol mittels des Feldes U: zugewiesen werden.

Das Feld besteht aus einem Großbuchstaben von H bis Y oder einem Kleinbuchstaben von h bis w, dann ein Gleichheitszeichen = und danach das Symbol, z.B.

```

U: T = !trill!
U: H = !fermata!
U: M = !tenuto!

```

Um eine Definition wieder zurückzusetzen, kann die folgende Definition benutzt werden:

```
U: T = !nil!
U: H = !nil!
U: M = !nil!
```

Die Buchstaben `uvTHLMPSO` sind vordefinierte Abkürzungen für häufige Symbole:

```
U      !upbow!
V      !downbow!
T      !trill!
H      !fermata!
L      !accent! or !emphasis!
M      !lowermordent!
P      !uppermordent!
S      !segno!
O      !coda!
```

Für Musikstücke, in denen eine große Anzahl von Symbolen verwendet wird, gibt es eine Alternative zum setzen der Symbole vor der jeweiligen Note. Nach einer Zeile Noten kann eine Zeile beginnend mit `d:` eingefügt werden. Diese Zeile enthält ausschließlich Symbole. Die Regeln für die Zuordnung von Symbolen zu Noten entsprechen denen für die Zuordnung von Text zu Noten, beschrieben in Kap. 2.9.

`d:`-Zeilen und den Noten direkt zugeordnete Symbole können gleichzeitig verwendet werden.

```
X: 1
L: 1/4
U: M = !accent!
K: C
C/D/ E/F/ G/A/ B/c/|c/B/ A/G/ F/E/ D/C/|e2!fermata!c2|
d: M M * !ff! M M * !ff! M M M !ff! M M M !ff!
```

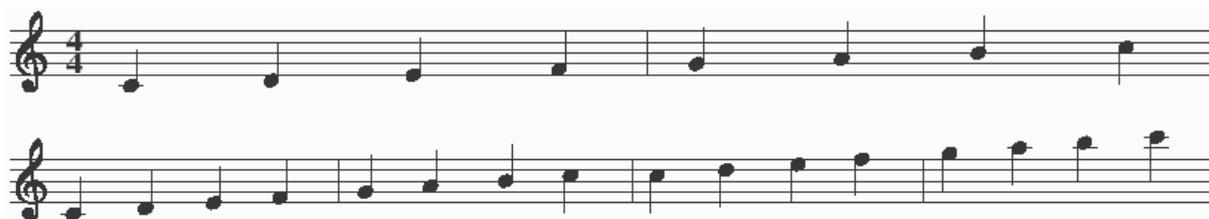


## 2.14 Zeilenumbrüche erzwingen: !

Mit dem Ausrufungszeichen `!` kann eine neue Notenzeile in der Ausgabe erzwungen werden.

Dann darf allerdings die `abcm2ps`-Option `-c` nicht benutzt werden. Im folgenden Beispiel werden zwei Notenzeilen erzeugt, die erste mit zwei Takten, die zweite mit vier.

```
X: 1
L: 1/4
K: C
CDEF|GABc|! CDEF|GABc|cdef|gabc'|
```



Die Verwendung von erzwungenen Zeilenumbrüchen sollte mit Vorsicht erfolgen, da die Ergebnisse die Lesbarkeit der Noten unter Umständen verschlechtern.

## 2.15 Zeilenumbrüche verhindern: \

Normalerweise ergeben n Takte im Quelltext n Takte in der Notenzeile der Ausgabedatei.

Manchmal ist es jedoch ungünstig, z.B. 6 Takte in einer Quelltextzeile zu schreiben, da dann wegen sehr langer Zeilen die Lesbarkeit des Quelltextes leidet.

In diesen Fällen kann das Backslash-Zeichen \ am Ende einer Quelltextzeile angefügt werden und der Rest der Notenzeile in einer neuen Quelltextzeile geschrieben werden.

Analog dazu kann auch eine Liedtext-Zeile auf mehrere Quelltextzeilen verteilt werden.

```
X: 1
T: Brother John
C: Traditional
L: 1/4
K: C
CDEC|CDEC|EFGz|\ % continues
w: Are you slee-ping, Are you slee-ping, Bro-ther John!\ % continues
EFGz|
w: Bro-ther John!
G/ A/ G/ F/ EC|G/ A/ G/ F/ EC|\ % continues
w: Mor-ning bells are rin-ging, Mor-ning bells are rin-ging,\
FB,Ez|FB,Ez|]
w: ding ding dong, ding ding dong!
```

### Brother John

*Traditional*

## 3 Gestaltung und Strukturierung

### 3.1 Notenschlüssel und Tonarten

Bis jetzt wurden alle Beispiele mit Violinschlüssel und in C-Dur geschrieben. Im Normalfall wird der passende Notenschlüssel von `abcm2ps` automatisch gewählt ausgehend von der Tonhöhe der verwendeten Noten. Der Bass-Schlüssel wird z.B. automatisch gewählt, wenn viele tiefe Noten geschrieben werden.

Der Notenschlüssel kann jedoch auch explizit über das `K`-Feld zu Beginn des Stückes gesetzt werden. Es besteht auch die Möglichkeit, auf einen Notenschlüssel gänzlich zu verzichten. Die verfügbaren Schlüssel und ihre Definitionen in ABC PLUS-Syntax sind in folgender Tabelle aufgeführt:

Schlüssel	Feld
ViolinSchlüssel	K: treble (default)
Violinschlüssel, 1 Oktave erniedrigt	K: treble-8
Violinschlüssel, 1 Oktave erhöht	K: treble+8
Bassschlüssel	K: bass
Bariton	K: bass3
Tenor	K: alto4
Alto	K: alto
Mezzosoprano	K: alto2
Soprano	K: alto1
Kein Notenschlüssel	K: none
Percussions	K: perc

Der Name mancher Schlüssel enthält eine Ziffer am Ende. Diese Ziffer gibt an, welche Notenlinie der Schlüssel umschließt.

Die Violinschlüssel mit `+8` oder `-8` im `K`-Feld werden mit einer kleinen 8 über bzw. unter dem Schlüssel dargestellt. Diese Notation wird oft für Chorstimmen des Tenor verwendet. Diese Stimme klingt eine Oktave tiefer als notiert.

Im nächsten Beispiel sind alle Schlüssel einmal dargestellt.

*Hinweis: `K`-Felder können auch innerhalb der Noten eingefügt werden. Man spricht von *Inline-Feldern*. Sie sind durch eckige Klammern eingeschlossen. Zwischen öffnender Klammer und *Feld-Typ-Zeichen* darf kein Leerzeichen stehen.*

```
X: 1
L: 1/4
K: none
CEGc | [K: C treble] CEGc |[K: Cm bass]cegc' |
w: none | treble | bass |
[K: C bass3]cegc' | [K: Cm alto4]CEGc| [K: C alto]cegc' |
w: baritone | tenor | alto |
[K: Cm alto2]cegc' | [K: C alto1]cegc' ||
w: mezzosoprano | soprano |
```

Neben dem Notenschlüssel wird über das K:-Feld die Tonart festgelegt. Es werden Großbuchstaben verwendet und ein kleines m angefügt, wenn es sich um Moll handelt. Vorzeichen werden als # bzw. b hinter dem ersten Buchstaben geschrieben. Wenn mit anderen Schlüsseln gearbeitet wird, kann abcm2ps die Musik automatisch um eine oder zwei Oktaven transponieren, um ein besseres Verhältnis zwischen Schlüssel und Noten zu erhalten. Zum Beispiel können beim Bass-Schlüssel die Noten c und C abhängig von der Situation equivalent sein. Damit wird verhindert, dass zu viele Kommas zu schreiben sind. Dies sei an folgenden Beispielen illustriert:

```
X: 1
T: Notes relative to "c"
L: 1/4
K: C bass
cdef|gabc'|CDEF|C,D,E,F,|
w: c d e f g a b c' C D E F C, D, E, F,

X: 2
T: Notes relative to "C,"
L: 1/4
K: C bass
C,D,E,F,|G,A,B,C|CDEF|cdef|
w: C, D, E, F, G, A, B, C C D E F c d e f
```

Notes relative to "c"

Notes relative to "C,"

*Hinweis: In der Theorie wäre es richtig, Bassnoten mit Kommas zu schreiben. In der Praxis ist es jedoch zu bevorzugen, diese Regel systematisch zu brechen...*

### 3.2 Taktart: M:

Mit dem M:-Feld kann die Taktart des Musikstückes auf verschiedene Art und Weise festgelegt werden:

- In gebrochener Darstellung, z.B. 3/4 oder 4/4. Komplexere Definitionen wie z.B. M: 5/4 (2/4 3/4) können benutzt werden.
- Als Ganzzahliger Wert
- In Textform : M: C bzw. M: C| entsprechend 4/4 und 2/4 ("alla breve")
- Wenn keine Taktangabe benötigt wird: M: none

Die Taktart eines Musikstückes kann mitten im Stück wechseln. In diesem Fall kann ein M:-Feld inline eingefügt werden.

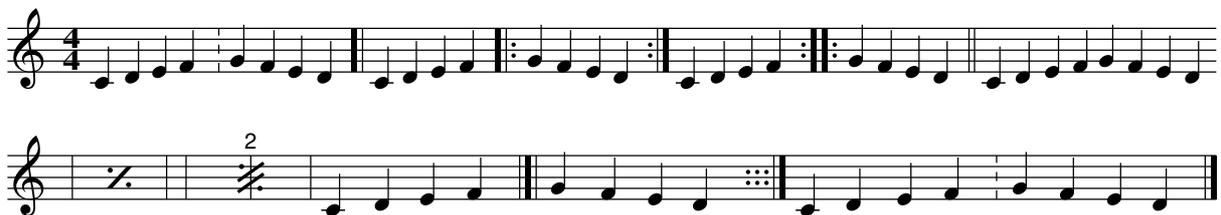
```
X: 1
M: C
K: C
L: 1/4
C D E F | G A B c | [M: 3/4] c d e|f g a| [M: 2/4] b c' |cG|EC|
```



### 3.3 Taktstriche und Wiederholungen

Zusätzlich zu normalen Taktstrichen können durch Kombination der Zeichen |, [, ], / und : Sonderformen des Taktstriches erzeugt werden.

```
X: 1
L: 1/4
K: C
CDEF : GFED [| CDEF |: GFED :|CDEF :: GFED [| CDEF [|] GFED
|/| |//| CDEF [|] GFED :::] CDEF .| GFED |]
```



Die Kombination [|] gibt keinen Taktstrich aus. Sie definiert einen unsichtbaren Taktstrich, der z.B. als Platzhalter für Symbole benutzt werden kann.

Um darzustellen, dass eine Wiederholung unterschiedliche Schlußsequenzen besitzt, können die Kombinationen [1 und [2 benutzt werden. Wenn ein Kasten mit einem Wiederholungszeichen beginnt kann statt [2 auch |2 geschrieben werden.

```
X: 1
L: 1/4
K: C
|: C D E F | G F E D | [1 C2 G2 :|2 C G C z |]
```



Abcm2ps unterstützt noch weitere Formen der Wiederholungsnotation. Zur Beschreibung der Kästen können nicht nur Ziffern, sondern auch Punkte, Kommas, Bindestriche und Text in Anführungszeichen verwendet werden:

```
X: 1
L: 1/4
K: C
|: C D E F | 1-3 c d e f :|4,5 C2 G2 :|"last time" C G C z |]
```



### 3.4 Titel, Komponist, Tempo: T:, C:, Q:

Dem Notenblatt fehlt bisher noch einiges.... Im nächsten Beispiel werden die Felder

T: Titel, Untertitel

C: Komponist

Q: Tempo

Eingeführt.

```
X: 1
T: Happy Birthday % title
T: (Tanti auguri a te) % subtitle
C: traditional % composer
C: (transcription Guido Gonzato)
M: 3/4
Q: "Allegro" 1/4 = 120 % tempo
K: C
C> C | D2C2F2 | E2-E z C> C | D2C2G2 | F2-F z C> C |
w: Hap-py birth-day to you,_ Hap-py birth-day to you,_ hap-py
c2A2F2 | E2D z _B> B | A2F2G2 | F6 |]
w: birth-day dear fel-low, hap-py birth-day to you!
```

# Happy Birthday

(Tanti auguri a te)

traditional  
(transcription Guido Gonzato)

**Allegro**

Hap - py birth - day to you, — Hap - py birth - day to you, — hap - py  
birth - day dear fel - low, hap - py birth - day to you!

Die textuelle Angabe im Q:-Feld („Allegro“) kann weggelassen werden.  
In Kap. 5.3 wird beschrieben, wie die Schriftart für den Titel gewechselt werden kann.

## 3.5 Parts: P:

Manche Musikstücke bestehen aus verschiedenen Teilen, die in mehreren Varianten wiederholt werden. Um die Reihenfolge zu spezifizieren, in der die einzelnen Teile zu spielen sind, wird das P:-Feld genutzt. Das Feld enthält die Part-Namen.

Im Kopf des Quelltextes gibt dieses Feld die Reihenfolge der Teile an, im Musikstück selber den Beginn eines Parts.

Wenn das P:-Feld im Kopf des Quelltextes benutzt wird, so kann statt mehrmaligen schreiben des Part-Namens hinter dem jeweiligen Part-Namen eine Zahl angegeben werden, die eine mehrmalige Wiederholung des Parts anzeigt. P : A3 ist das gleiche wie P : AAA; P : (AB) 3C2 ist äquivalent zu P : ABABABCC. Um den Text lesbarer zu gestalten können Punkte zwischen den Part-Namen benutzt werden.

Ein etwas komplizierteres Beispiel soll dieses Kapitel beenden: P : ( (AB) 3 . (CD) 3 ) 2 ist identisch mit P : ABABABCD CDCDABABABCD CDCD.

Auf der nächsten Seite ist ein einfacheres Notenbeispiel zur Definition von Parts dargestellt.

```
X: 1
T: Song in three parts
L: 1/4
P: AABBC % or: P: A2.B2.C
K: C
[P: A] |: C D E F|C D E F|G G G G|G2 z2:|:
[P: B] C E G c|C E G c|c c c c|c2 Cz:|
[P: C] C/E/G/c/ C2|C/E/G/c/ C2|C4|]
```

### Song in three parts

AABBC

A B

C

### 3.6 Begleitakkorde

In vielen Liederbüchern werden Begleitakkorde, z.B. für Gitarre, über dem Notensystem angegeben. In ABC PLUS werden solche Akkordnamen in Anführungszeichen direkt vor die Note gesetzt, ab der sie gelten.

Ein Begleitakkord hat folgendes Format:

```
<Note> [Vorzeichen] [Typ] [/Bass Note]
```

Note steht für A...G (nur Großbuchstaben), als Vorzeichen kann b oder # verwendet werden, als Typ sollten die in nachfolgender Tabelle angegebenen Akkordtypen verwendet werden. Optional kann ein Slash / gefolgt von einer Bassnote A...G angegeben werden. Da die ABC-Notation aus dem englischsprachigen Raum stammt und es dort die Note H als B bezeichnet wird, sollte die Notation entsprechend erfolgen, zumindest wenn abc2MIDI verwendet werden soll. Leerzeichen zwischen Akkord und Note sind nicht erlaubt.

Verschiedene Typen von Begleitakkorden:

Type	Meaning
m oder min	Moll
maj	Dur
dim	vermindert
+ oder aug	übermäßig
sus	sustained
7, 9, ...	Sept, Nonen, ...

```
X: 1
T: Happy Birthday
T: (version with chords)
C: traditional
M: 3/4
Q: "Allegro" 1/4 = 120 % tempo
K: C
C> C|"F"D2C2F2|"C"E3 z C> C|"C"D2C2G2|
w: Hap-py birth-day to you, Hap-py birth-day to
"F"F3 z C> C|"F"c2A2F2|"Bb"E2D z _B> B|
w: you, yap-py birth-day dear fel-low, hap-py
"F"A2F2"C"G2|"F"F6||
w: birth-day to you!
```

#### Happy Birthday (version with chords)

*traditional*

**Allegro**

Hap - py birth - day to you, Hap - py birth - day to you, yap - py birth - day dear  
fel - low, hap - py birth - day to you!

### 3.7 Textergänzungen: “^\_<>@”

Textuelle Zusätze zu den Noten können auf verschiedene Art und Weise erzeugt werden.

Eine Möglichkeit ist es, den Text als Begleitakkord zu schreiben, d.h. in Anführungszeichen vor der Note, zu der der Text in Beziehung steht.

Nach dem ersten Anführungszeichen sollte zunächst ein spezielles Zeichen (^<>oder @) folgen um Verwechslungen mit der Definition von Begleitakkorden zu vermeiden.

Eine andere Methode zum Einfügen von Text ist die Verwendung des P:-Feldes. Schliesslich können auch Q:-Felder benutzt werden, um Tempoänderungen anzuzeigen.

Textdefinitionen sollten mit einem der Zeichen ^\_<>@ beginnen. Diese Zeichen geben an, wo der Text platziert werden soll:

^ über dem Notensystem

\_ unter dem Notensystem

< links der Note, vor der der Text definiert ist

> rechts der Note

@ dieses Zeichen dient dazu, Text relativ zum Notenkopf an beliebigen Stellen zu positionieren. Wenn es sich um einen Akkord handelt, dann relativ zur tiefsten Note. Der Offset wird in Punkten angegeben, für X und Y getrennt durch ein Komma

Im folgenden ein Beispiel, welches alle Methoden einmal verwendet:

```
X: 1
Q: "Dolcemente" 1/4=60
L: 1/4
K: C
CDEF| [P:piano]GFED| "^above"CDEF| "_below"GFED| "<left "c' ">right"A, DE|
[Q: "sostenuto"] FGC"@-15, 5.7anywhere"D|
```

Dolcemente ♩ = 60

piano above below left right anywhere

sostenuto

### 3.8 Informationsfelder

In Kap. 1.7 wurde bereits erwähnt, daß eine ABC-Datei mehrere Musikstücke enthalten kann. Diese Möglichkeit und die Einfachheit der Notation führte zur Erstellung von vielen im Internet verfügbaren Musik-Archiven. In der Tat wurde ABC ein Standard für die Verbreitung von Musik, insbesondere von Folk-Musik und Traditionals.

Es gibt Feld-Definitionen, die dazu dienen, Einordnungskriterien für ein Musikstück zu definieren. Diese Angaben erscheinen nicht in der Notenausgabe, können aber zur Suche nach bestimmter Musik in Musikarchiven benutzt werden.

Es gibt folgende Felder:

<b>A:</b> Ort (engl. Area).	Example: A: Northern Italy
<b>B:</b> Buch	Example: B: The Beatles 66-72
<b>D:</b> Diskographie	Example: D: Ricordi
<b>F:</b> Dateiname	Example: F: <a href="http://www.mymusic.org/file.abc">http://www.mymusic.org/file.abc</a>
<b>G:</b> Gruppe	Example: G: guitar
<b>H:</b> Geschichte (engl. History)	Example: H: this tune was written in...
<b>I:</b> Information	Example: I: version per single voice
<b>N:</b> Notizen	Example: N: see also...
<b>O:</b> Original	Example: O: English
<b>R:</b> Rhythmus.	Example: R: Waltz
<b>S:</b> Quelle (engl. Source)	Example: S: collected near Verona
<b>Z:</b> Transcriptions Notizen	Example: Z: from photocopy

## 4 Teil III Harmonie - Polyphonie in ABC PLUS

Bis jetzt wurde der Umgang mit Musikstücken beschrieben, die für eine einzelne Stimme oder ein einzelnes Instrument geschrieben wurden.

Dies ist der Funktionsumfang der herkömmlichen ABC-Notation.

In diesem Teil werden nun die Erweiterungen von ABC PLUS hinsichtlich polyphoner Musik beschrieben. Es werden vorwiegend Chorsätze als Beispiele verwendet.

### 4.1 Stimmen und Notensysteme

Zunächst ein kleiner Ausflug in die Musiktheorie. In einer Partitur sind jeweils mehrere Notenzeilen zusammengefasst, d.h. sie besteht aus mehreren Stimmen. Die Stimmen beziehen sich auf einzelne Instrumente, welche wiederum einstimmig (z.B. Blasinstrumente) oder mehrstimmig (z.B. Klavier, Orgel) sein können.

Ein Satz aus zusammengefassten Notenzeilen wird im folgenden als Notensystem bezeichnet.

*Hinweis: abcm2ps unterstützt bis zu 16 Stimmen. Diese Begrenzung kann jedoch aufgehoben werden, wenn die abcm2ps-Quelltexte angepasst und neu übersetzt werden.*

Als erstes Beispiel folgt nun ein Stück mit zwei Stimmen, wobei jede Stimme auf einer eigenen Notenzeile notiert wird. Das V:-Feld, gefolgt vom Namen der Stimme, gibt an, dass die folgende Notenzeile zu dieser Stimme gehört. Als Name kann eine Zahl oder eine Zeichenkette (z.B. "Tenor") verwendet werden.

Das V:-Feld kann in einer eigenen Zeile notiert werden, oder in eckige Klammern eingeschlossen zu Beginn einer Notenzeile.

```
X: 1
T: Brother John
C: Traditional
L: 1/4
K:E
V: 1
EFGE | EFGE | GABz | GABz | B/c/B/A/ GE | B/c/B/A/ GE |
V: 2
z4 | z4 | EFGE | EFGE | GABz | GABz |
V: 1
FB, Ez | FB, Ez | z4 | z4 |
V: 2
B/c/B/A/ GE | B/c/B/A/ GE | FB, Ez | FB, Ez |
```

## Brother John

*Traditional*

In diesem Beispiel wurde abwechselnd Zeile 1 und 2 des Stückes notiert, so wie die Noten auch in der Ausgabe erscheinen. Es hätte auch zuerst die komplette Stimme 1 und danach Stimme 2 notiert werden können.

Im Kopf des Quelltextes können zu jeder Stimme weitere Angaben eingebracht werden.

```
V: <Stimmen Name> [Definitionen]
```

Der Name der Stimme kann eine Zahl oder ein Wort sein. Mögliche Definitionen sind:

- **clef=** spezifiziert den Notenschlüssel für die Stimme. Es können die in Kap. 3.1 angegebenen Definitionen verwendet werden.
- **name=** <name> oder **nm=** <name> spezifiziert den Namen der links der ersten Notenzeile erscheint.
- **sname=** <nam > oder **snm=** <name> spezifiziert den Namen der links ab der zweiten Notenzeile der Stimme erscheint.
- **merge** zeigt an, dass diese Stimme in der gleichen Notenzeile ausgegeben wird wie die vorige Stimme.
- **up and down** geben die Richtung der Notenhälsa an (up: nach oben, down: nach unten).

Alle Definitionen sind optional. Als Beispiel nochmals das gleiche Lied mit einigen Verbesserungen:

```
X: 1
T: Brother John
C: Traditional
L: 1/4
V: 1 clef=treble name="Contralto" sname="S"
V: 2 clef=treble name="Voce bianca" sname="VB"
K: E
%
[V: 1] EFGE|EFGE|GABz|GABz|B/c/B/A/ GE|B/c/B/A/ GE|
[V: 2] z4 |z4 |EFGE|EFGE|GABz |GABz |
%
[V: 1] FB,Ez |FB,Ez |z4 |z4 |
[V: 2] B/c/B/A/ GE|B/c/B/A/ GE|FB,Ez|FB,Ez|
```

## Brother John

Traditional

Contralto

Voce bianca

S

VB

! Es gibt noch eine dritte Möglichkeit, V:-Felder anzugeben. Es kann auch zu Beginn einer Notenzeile ohne eckige Klammern geschrieben werden:

```
V:1 CDEF|GABc|
```

Diese Notation sollte jedoch nicht benutzt werden, da das Programm abc2MIDI und andere Applikationen dieses V:-Feld als Fehler ansehen!

### 4.2 Notenzeilen positionieren: %%staves

Es gibt Instrumente, deren Stimme in mehr als einer Notenzeile notiert wird (Klavier, Orgel, ...).

Andererseits ist es teilweise üblich, mehrere Stimmen in einer Notenzeile zusammenzufassen. Um zu definieren, wie die einzelnen Stimmen auf die Notenzeilen des Systems verteilt werden, wird die %%staves Anweisung benutzt.

Hinter der %%staves Anweisung werden die Namen der Stimmen notiert, optional von Begrenzungszeichen umfasst: [], {}, und ().

Wie andere Anweisungen im Kopf des Quelltextes muss %%staves vor dem K:-Feld stehen. Wenn in den Noten eine Stimme verwendet wird, die nicht im Kopf deklariert wurde, wird sie ignoriert.

*Hinweis: Die %%staves Anweisung beginnt mit einem %, so dass sie eigentlich als Kommentar gilt und ignoriert werden sollte. Dies ist jedoch nicht der Fall. Wenn ein Kommentar mit %% beginnt, so wird er Meta-Kommentar genannt. Dies wurde eingeführt, um die Kompatibilität zu ABC zu wahren. Der zusätzliche Leistungsumfang von ABC PLUS wird weitgehend über diese Art von Anweisungen gesteuert. Programme, die diese Anweisungen nicht kennen, ignorieren sie einfach als Kommentar und arbeiten trotzdem erfolgreich.*

Die Begrenzungszeichen werden nach folgenden Regeln verwendet:

- Wenn die Stimmen nicht von Begrenzungszeichen eingeschlossen sind, werden sie in separaten Notenzeilen ausgegeben. Die oberste Stimme im System muss die erste Stimme in der Liste sein. Beispiel: %%staves SATB

- Wenn zwei oder mehr Stimmen in eckige Klammern eingeschlossen werden, dann werden die zugehörigen Notenzeilen mit einem dicken Balken als zusammengehörig dargestellt. Diese Form der Darstellung wird oft bei Chorsätzen verwendet. Beispiel: `%%staves [SATB]`.
- Werden zwei oder mehr Stimmen in geschweifte Klammern eingeschlossen, so werden die zugehörigen Notenzeilen mit geschweiften Klammern zusammengefasst. Diese Notierung wird meist für Klavier und Orgel verwendet. Beispiel: `%%staves { MS MD }`
- Runde Klammern um zwei oder mehr Stimmen bewirken, dass diese Stimmen in die gleiche Notenzeile ausgegeben werden. Beispiel: `%%staves [(SA) (TB)]`
- Taktstriche werden normalerweise über alle Notenzeilen des Systems gezogen, auch durch die Zwischenräume zwischen den Zeilen. Um dies zu unterbinden kann das Zeichen `|` zwischen den Stimmnamen benutzt werden: `%%staves [S|A|T|B]`
- Wenn zwei Stimmen in der gleichen Notenzeile ausgegeben werden sollen, wird durch die Richtung der Notenhäule angegeben, welches die erste Stimme ist (Notenhäule nach oben) und welches die zweite Stimme ist (Notenhäule nach unten).

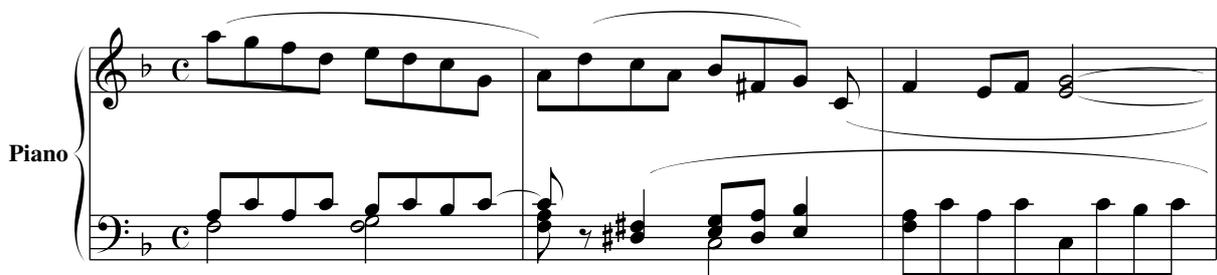
Als Beispiel folgt nun ein Klavierstück. Es besteht aus drei Stimmen, wobei zwei Stimmen von der linken Hand gespielt werden sollen.

Wenn eine der Stimmen nichts zu spielen hat, werden die normalen Pausen durch unsichtbare Pausen ersetzt (siehe Kap. 2.3).

```
X:1
T: Studio
T: Op. 10 - N. 3
C: F. Chopin
M: C
%%staves {RH1 (LH1 LH2)}
V: RH1 clef=treble name="Piano"
V: LH1 clef=bass
V: LH2 clef=bass
K: F
%
[V: RH1] (agfd edcG |A) (dcA B^FG) (C |F2 EF [E4G4]- |
[V: LH1] ac'ac' bc'bc' -|c' z ([^d2^f2][eg][da][e2b2] |[fa]c'ac'cc'bc' |
[V: LH2] f4 [f4g4] |[fa] x x2 c4 |x4 x4 |
%
```

Studio  
Op. 10 – N. 3

F. Chopin



Nun ein etwas komplexeres Beispiel. Es handelt sich um die ersten 4 Takte aus Mozart's "Ave Verum" für Orgel und Chor:

```

X: 1
T: Ave Verum
C: W. A. Mozart
M: 4/4
L: 1/4
Q: "Adagio"
%%staves [(S A) (T B)] {(MD1 MD2) (MS1 MS2)}
V: S clef=treble name="Soprano" sname="S"
V: A clef=treble name="Alto" sname="A"
V: T clef=bass name="Tenore" sname="T"
V: B clef=bass name="Basso" sname="B"
V: MD1 clef=treble name="Organo"
V: MD2 clef=treble
V: MS1 clef=bass
V: MS2 clef=bass
K: D
%
[V: MD1] (DA,D[CE]) | ([DF]D[DF] [EG]) | [FA] [DF] [Fd] [DF] | A^G=GG |
[V: MD2] x4 |x4 |x4 |E4 |
[V: MS1] f2fa |afa2- |a4 |b4 |
[V: MS2] d4- |d4- |d4- |d4 |
[V: B] z4 |z4 |d2d2 |d2d2 |
w: A- ve, A- ve,
[V: T] z4 |z4 |a2a2 |b2b2 |
[V: A] z4 |z4 |F2F2 |E2E2 |
[V: S] z4 |z4 |A2 (dF) | (A^G)=G2 |
w: A- ve, * A - ve,

```

## Ave Verum

W. A. Mozart

**Adagio**

Zu beachten ist, dass die Stimmen absichtlich in der verkehrten Reihenfolge im Quelltext notiert sind. Die %%staves Anweisung sorgt dafür, daß die Notenzeilen und Stimmen in der richtigen Reihenfolge dargestellt werden. Normalerweise würde man die einzelnen Stimmen natürlich in der Reihenfolge im Quelltext notieren, in der sie auch dargestellt werden.

Hinweis: Die %%staves Anweisung ist ein großer Vorteil von ABC PLUS gegenüber Programmen mit graphischer Oberfläche. Zum Beispiel bedarf es nur einer Änderung in der %%staves-Anweisung, um aus einem vierstimmigen Notensatz mit einer Notenzeile pro

Stimme einen Notensatz mit zwei Stimmen pro Notenzeile zu erzeugen. Mit den meisten graphischen Programmen müsste das Stück komplett neu geschrieben werden.

Es sollte im allgemeinen bevorzugt werden, die Stimmen in der Reihenfolge zu notieren, in der sie im Notenblatt dargestellt werden.

Als letztes Beispiel folgt nun ein Musikstück, welches in einer ungewöhnlichen Art und Weise geschrieben ist: "Kyrie" von Andrea Gabrieli's "Missa Brevis". Diese Musik hat keine Taktart und jede Stimme folgt ihrem eigenen Tempo: In dieser Situation muss M:none benutzt werden. Die Länge der Takte ist unterschiedlich für jede Stimme, deshalb werden die Taktstriche durch das !longphrase! Symbol ersetzt. Außerdem soll als Tempoangabe "alla breve" angezeigt werden.

```
X: 1
T: Missa Brevis
C: Andrea Gabrieli (1510? - 1586)
M: C|
L: 1/4
%%staves [1 2 3 4]
V: 1 clef=treble
V: 2 clef=treble
V: 3 clef=treble-8
V: 4 clef=bass
U: L = !longphrase!
K: F
%
[P: Kyrie]
[V: 1] [M:none] F4 c2d2c2LG2 A2B2c2A2G2LF2 G2 c4 =B2 Lc4 z2 G2
w: Ky- ri - e e- lei - - - son e- lei - - son Ky-
[V: 2] [M:none] Lz8 C4 F2G2 FECD E2 F4 E2C2G2A2G2F2E2
w: Ky- ri - e * * e- lei - - son e- lei - - -
[V: 3] [M:none] z8 Lz8 F4 c2d2c2G2A2d2f2e2d2c2
w: Ky- ri - e e- lei - - - - -
[V: 4] [M:none] z8 z8 Lz8 c4 f2g2f2Lc2 d2e2
w: Ky- ri - e e- lei -%
[V: 1] c2d2c2LG2 A2B2A3 GAB c2 d4 c3 B/LA/ G4 A16 |]
w: ri - e e- lei - - - - - son.
[V: 2] A2 F4 E2F2D2 F4 F2 G3 F LF2 E2 F4 E2 F16 |]
w: - - - son Ky- ri- e~e- lei - - - - - son.
[V: 3] A3 =B Lc4 z2 G2c2d2c2LG2 A2_B2G2LA2 c4 c16 |]
w: son__ Ky- ri - e e- lei - - - - son.
[V: 4] Lf4 z2 c2f2g2f2Ld2 f2e2d2LB2 c8 f16 |]
w: son Ky- ri - e e- lei - - - - son.
```

# Missa Brevis

Andrea Gabrieli (1510? – 1586)

## Kyrie

Ky - ri - - e e - lei - - - son e - lei - - son

Ky - - ri - - e e - lei - - son e - lei - -

Ky - - ri - - e e - lei - -

Ky - - ri - - e e - lei

Ky - ri - - e e - lei - - - - - son.

- - - son Ky - ri - - e e - lei - - - - - son.

- son Ky - ri - - e e - lei - - son.

- son Ky - ri - - e e - lei - - - - - son.

### 4.3 Stimmteilung

In einigen Musikstücken teilt sich eine Stimme nur für einige Takte in zwei Stimmen auf. Um die Einführung einer kompletten, überwiegend indentschen Stimme zu vermeiden, kann das &-Zeichen benutzt werden. Wenn es innerhalb eines Taktes geschrieben wird, so teilt es die aktuelle Stimme und weist die folgenden Noten der zweiten Stimme zu.

```
X: 1
L: 1/4
K: C
C>CE>E|G>GG2 & G2E2|C>CE>E|G>GG2 & x2E2|
```

So kann dieses Stück alternativ ohne Stimmteilung, aber mit zwei Stimmen von Beginn an geschrieben werden:

```
X: 1
L: 1/4
%%staves (1 2)
K: C
[V:1] C>CE>E|G>GG2|C>CE>E|G>Gc2|
[V:2] x4 |G2E2 |x4 |x2E2 |
```

## 4.4 Wechsel des Notensystems

In komplexeren Musikstücken (z.B. für Solist, Chor und Orchester) spielen nicht alle Instrumente zur gleichen Zeit. Große Teile des Stückes nur mit Pausen zu füllen, kostet unnötig Platz, wenn nur ein Instrument spielt.

Das `%%staves` Feld kann entsprechend den aktuellen Gegebenheiten geändert werden. Es müssen nur die Stimmen angegeben werden, welche spielen.

Als Beispiel sei hier "Riu riu chiu" angeführt, ein bekanntes Stück von Villancico aus dem 16. Jh.

Im Titel kommt die Zeichensequenz `\241` vor. Das ist der Oktal-Code für das Zeichen `j` im Zeichensatz ISO 8859-1 (Latin1).

```
%%scale 0.68
%%barsperstaff 6
X: 1
T: Riu, riu, chiu, \241la guarda ribera!
C: Villancico (Spagna, XVIth century)
M: C|
L: 1/2
Q: 1/2 = 240
%%staves 3
V: 3 clef=treble-8 name="Tenor\nBass"
K: Am
% MEN ONLY
[V: 3] [M:none] AAGA|F2ED2EFG|A2A2|
w: Ri-u, ri-u, chi-u, \241la guar-da ri-be-ra!
[V: 3] AAGA |F2EG2GEF|D2D2|
w: Di\'os guar-d\'o el lo-bo de nue-stra cor-de-ra,
[V: 3] AAGA |F2EG2GEF|D2D2|
w: Di\'os guar-d\'o el lo-bo de nue-stra cor-de-ra.
% SYSTEM CHANGE: ALL
%%staves [1 2 3 4]
V: 1 clef=treble name="S" sname="S"
V: 2 clef=treble name="A" sname="A"
V: 3 clef=treble-8 name="T" sname="T"
V: 4 clef=bass name="B" sname="B"
[V: 1]AAGA|F2ED2EFG |A2A2z2|
w: Ri-u, ri-u, chi-u, la guar-da ri-be-ra!
[V: 2]FFEC|D2EF2EDD |C2C2z2|
[V: 3]cccG|A2AA2ADD |E2E2z2|
w: Ri-u, ri-u, chi-u, la guar-da ri-be-ra!
[V: 4]ffcf|d2Ad2c_BB|A2A2z2|
%
[V: 1] z4 |AAGA|F2EF2FEE|D2D2z2|
w: Di\'os guar-d\'o el lo-bo de nue-stra cor-de-ra,
[V: 2] z2EE |DCEC|D2CD2DCC|D2D2z2|
w: Di\'os guar-d\'o el lob', el lo-bo de nue-stra cor-de-ra,
[V: 3] ccBc |A2BA|A2AA2AAA|A2A2z2|
w: Di\'os guar-d\'o el lo-bo, el lo-bo de nue-stra cor-de-ra,
[V: 4] aaga |f2ef|d2Ad2dAA|d2d2z2|
%
[V: 1] z4 |AAGA|F2ED2DCC |D2D2 |
w: Di\'os guar-d\'o el lo-bo de nue-stra cor-de-ra.
[V: 2] z2EE|DCEC|D2CA,2A,A,A,|A,2A,2|
w: Di\'os guar-d\'o el lob', el lo-bo de nue-stra cor-de-ra.
```

```

[V: 3] ccBc|A2BA|A2AF2FEE |D2D2 |
w: Di\'os guar-d\'o el lo-bo, el lo-bo de nue-stra cor-de-ra.
[V: 4] aaga|f2ef|d2Ad2dAA |d2d2 |
% SYSTEM CHANGE: MEN ONLY
%%staves 3
[V: 3] AAGA |F2EG2GEF|D4 |AAGA |
w: El lo-bo ra-bio-so la qui-so mor-der, Mas Di\'os po-de-
[V: 3] F2FEGGEF|D4 |AAGA |F2FEDEFG|
w: ro-so la su-po de-fen-der; qui so-le ha-ce que no pu-die-sce pe-
[V: 3] A4 |AAGA |F2FEGGEF|D2D2 |
w: car: ni~aun o-ri-gi-nal e-sta Vir-gen no tu-vie-ra.

```

Eine Alternative, um die Noten für dieses Stück zu schreiben, ist, drei einzelne Musikstücke in einer Datei zusammenzufassen, wobei jedes einer der drei Systemaufteilungen entspricht. Man kann dies tun, hat jedoch den Nachteil, dass eine Umwandlung nach MIDI auch drei Dateien liefert anstatt einer.

# Riu, riu, chiu, ¡la guarda ribera!

Villancico (Spagna, XVIth century)

Tenor  
Bass

8 Ri - u, ri - u, chi - u, ¡la guar - da ri - be - ra! Dí'os guar - d'ó el lo - bo de nue - stra cor - de - ra,

8 Dí'os guar - d'ó el lo - - bo de nue - stra cor - - de - - ra.

S  
A  
T  
B

S: Ri - u, ri - u, chi - u, la guar - da ri - be - ra! Dí'os guar - d'ó el lo - bo de nue - stra cor -  
A: Dí'os guar - d'ó el lob', el lo - bo de nue - stra cor -  
T: 8 Ri - u, ri - u, chi - u, la guar - da ri - be - ra! Dí'os guar - d'ó el lo - - bo, el lo - bo de nue - stra cor -  
B:

S  
A  
T  
B

S: de - ra, Dí'os guar - d'ó el lo - bo de nue - stra cor - de - ra.  
A: de - ra, Dí'os guar - d'ó el lob', el lo - bo de nue - stra cor - de - ra.  
T: 8 de - ra, Dí'os guar - d'ó el lo - - bo, el lo - bo de nue - stra cor - de - ra.  
B:

T

8 El lo - bo ra - bio - so la qui - so mor - der, Mas Dí'os po - de - ro - so la su - po de - fen - der;

T

8 qui so - le ha - ce que no pu - die - sce pe - car: ní'aun o - ri - gi - nal e - sta Vir - gen no tu - vie - ra.

## 5 Teil IV Seitenlayout

### 5.1 Formatierungsparameter

In den vorangegangenen Kapiteln haben wir gelernt mehrstimmige Musik zu schreiben. Jetzt soll es um das Seitenlayout, Schriftarten usw. gehen.

Abcm2ps besitzt verschiedene Parameter zur Beeinflussung der Formatierung. Diese Parameter werden im Quelltext als Meta-Kommentare geschrieben oder alternativ in Format-Dateien.

Meta-Kommentare (von jetzt an als Kommandos bezeichnet) sind Zeilen, die mit `%%` beginnen wie z.B. `%%staves`. Die Kommandos können im Kopf oder Rumpf des Quelltext stehen.

Es gibt verschiedenste Kommandos: für das Seitenlayout, Schriftarten, Abstände usw..

Viele Kommandos benötigen einen der folgenden Parameter

- Eine Längeneinheit, angegeben in Zentimeter (cm), Zoll (in) oder Punkten (pt), zB. 30pt, 1cm oder 0.3in
- Einen logischen Wert 'yes' or 'no'
- Einen Text wie z.B. Times Roman 24
- Eine Zahl, entweder ganzzahlig (Integer) oder gebrochen (real, mit einem Punkt als Komma und Nachkommastellen).

Im folgenden soll nun ein Beispiel betrachtet werden. Das Stück – die ersten 10 Takte aus Mozarts Ave Verum – enthalten die am meisten verwendeten Kommandos:

```
% PAGE LAYOUT
%
%%pageheight 29.7cm
%%pagewidth 21cm
%%topmargin 1cm
%%botmargin 1cm
%%leftmargin 1cm
%%rightmargin 1cm
% SPACING
%%topspace 0cm % space before the piece
%%titlespace 0cm % space before the title
%%subtitlespace 0.2cm % space before the subtitle
%%composerspace 0.5cm % space before the composer line
%%musicospace 0.5cm % space before the first staff
%%vocalspace 1.5cm % additional space after lyrics lines
%%sysstaffsep 1cm % space between staves in the same system
%%staffsep 2cm % space between different systems
% FONT
%%titlefont Times-Bold 32
%%subtitlefont Times-Bold 24
%%composerfont Times-Italics 16
%%vocalfont Times-Roman 14 % for lyrics
%%gchordfont Times-Bold 14 % for chords
% MISC
%%measurebox true % measure numbers in a box
%%measurenb 0 % measure numbers at first measure
%%exprabove true % expressions above the staff
%%barsperstaff 5 % number of measures per staff
%%scale 0.7 % magnification
%
X: 1
T: Ave Verum
T: per coro e organo
C: W. A. Mozart (1756-1791)
```

```

M: 4/4
L: 1/4
Q: "Adagio"
%%staves [(1 2) (3 4)] {(5 6) (7 8)}
V: 1 clef=treble name="Soprano" sname="S"
V: 2 clef=treble name="Alto" sname="A"
V: 3 clef=bass name="Tenore" sname="T"
V: 4 clef=bass name="Basso" sname="B"
V: 5 clef=treble name="Organo"
V: 6 clef=treble
V: 7 clef=bass
V: 8 clef=bass
K: D
% 1 - 5
[V: 1] z4 |z4 |A2 (dF) | (A^G)=G2 | (GB) (AG) |
w: A- ve, _ a - ve, ve - rum_
[V: 2] z4 |z4 |F2F2 |E2E2 | (EG) (FE) |
[V: 3] z4 |z4 |a2a2 |b2b2 |a2a2 |
w: A- ve, a- ve, ve- rum
[V: 4] z4 |z4 |d2d2 |d2d2 |c2c2 |
[V: 5] (DA,D[CE]) | ([DF]D[DF] [EG]) | [FA] [DF] [Fd] [DF] |A^G=GG | [EG] [GB] [FA] [EG] |
[V: 6] x4 |x4 |x4 |E4 |x4 |
[V: 7] f2fa |afa2- |a4 |b4 |a4 |
[V: 8] d4- |d4- |d4- |d4 |c4 |
% 6 - 10
[V: 1] (GF)F2 |E3E |FFGG | (G2F)F |E4 |
w: cor - pus na- tum de Ma- ri- a Vir - gi- ne,
[V: 2] (ED)D2 |C3C |DDEE | (E2D)D |C4 |
[V: 3] a2a2 |a3a |aaaa |a3a |a4 |
w: cor- pus na- tum de Ma- ri- a Vir- gi- ne,
[V: 4] d2d2 |A3A |ddcc |d3d |A4 |
[V: 5] [EG] [DF] [DF] [FA] |AEEA | [FA] [df] [eg]G | [E2G2] [D2F2] | [C4E4] |
[V: 6] x4 |C2C2 |DAAD |x4 |x4 |
[V: 7] a4 |a4 |a3a |a4 |x4 |
[V: 8] d4 |A4 |d2c2 |ddfd |Aaec |

```

# Ave Verum

## per coro e organo

W. A. Mozart (1756–1791)

Adagio

The first system of the musical score is for the Soprano and Alto voices, Tenore and Basso voices, and the Organ. The tempo is marked 'Adagio'. The key signature is one sharp (F#) and the time signature is 4/4. The lyrics for the Soprano and Alto parts are: 'A - ve, a - - ve, ve - rum cor - pus'. The Tenore and Basso parts have the same lyrics. The organ part features a melodic line in the right hand and a harmonic accompaniment in the left hand.

The second system of the musical score continues the vocal and organ parts. The lyrics for the Soprano and Alto parts are: 'na - - - tum de Ma - ri - a Vir - - - gi - ne,'. The Tenore and Basso parts have the same lyrics. The organ part continues with its melodic and harmonic accompaniment.

## 5.2 Änderung von Parametern

Wenn die Parameter einmal gesetzt wurden, gelten sie meist für das gesamte Stück. Einige Parameter können wiederholt gesetzt werden. Ein Beispiel dafür ist `%%vocalfont`. Damit kann die Schriftart für den Liedtext mehrfach gewechselt werden. Dies kann sinnvoll sein, wenn der Liedtext in mehreren Sprachen angegeben wird und für jede Sprache eine eigene Schriftart verwendet wird:

```
X: 1
T: Silent Night
C: F. Gruber
M: 3/4
Q: "Andante tranquillo"
K: C
%
G>A G E3|G>A G E3|d2 d B2 B|c2 c G3|
%%vocalfont Times-Roman 12
w: A- stro del ciel, Par- gol di- vin, \
```

```
w: mi- te~A- gnel- lo re- den- tor!
%%vocalfont Times-Italic 12
w: Voi- ci No- \^el, \^o dou- ce nuit! \
w: L'\^e- toile~est l\^a qui nous con- duit.
%%vocalfont Times-Roman 12
w: Si - lent night! Ho - ly night! All is calm, all is bright.
```

## Silent Night

F. Gruber

**Andante tranquillo**

A - stro del ciel, Par - gol di - vin, mi - te A - gnel - lo re - den - tor!  
 Voi - ci No - ël, ô dou - ce nuit! L'é - toile est là qui nous con - duit.  
 Si - - lent night! Ho - - ly night! All is calm, all is bright.

Diese Methode kann auch benutzt werden, um die Schriftart innerhalb einer Zeile zu wechseln:

```
%%font Helvetica
%%font Helvetica-BoldOblique
X: 1
L: 1/4
K: C
CDEF|!ff!GAB!fermata!c|!mf!cBAG|!p!FED!fermata!C|
%%vocalfont Helvetica 12
w: la la la la\
%%vocalfont Helvetica-BoldOblique 13
w: la la la la, la la la la\
%%vocalfont Helvetica 12
w: la la la la.
```

la la la la *ff* la la la la, *mf* la la la la *p* la la la la.

### 5.3 Verwendung von Schriftarten

Abcm2ps unterstützt nahezu alle POSTSCRIPT Schriftarten welche in Anhang E gelistet werden. Drei Schriftarten sind besonders wichtig: Times, Helvetica und Courier; alle mit kursiven und fetten Varianten. Times ist äquivalent zu der Windows Schriftart Times New Roman, Helvetica ist äquivalent zu Arial, und Courier zu Courier New.

Diese Schriftarten können als vordefinierte Schriftarten jederzeit mit jedem font-Kommando benutzt werden. Um weitere Schriftarten nutzen zu können, müssen diese mit einem %%font Kommando im Kopfteil des Quelltextes deklariert werden. Das folgende Beispiel demonstriert die abwechselnde Benutzung verschiedener Schriftarten in einem Musikstück.

```

% declare non-predefined fonts
%%font AvantGarde-Book
%%font Bookman-Light
%
%%titlefont Times-Italic 21
%%musicspace -0.5cm
%%textfont Helvetica 26
%%center Typesetting example
%%vskip 0.4cm
%%textfont Bookman-Light 14
%%begintext justify
%%This is an example of text inserted into an ABC file. This feature of
%%abcm2ps allows for the writing of songbooks, music collections or other
%%publications without having to resort to a word processor. Not bad, is
%%it? Now let's write a brief musical example.
%%endtext
X: 1
T: Etude
M: 4/4
L: 1/4
Q: "Dolcemente"
K: C
%
!p!CCGG|AA!mf!G2|!diminuendo(!FFEE|DD!diminuendo)!C2|
%%vskip 0.4cm
%%textfont AvantGarde-Book 14
%%begintext align
%%Now we'll have a look at something more lively. To start with, let's
%%switch fonts: from Bookman-Light to AvantGarde-Book. Here is the same
%%Etude with a few small varations to make it more interesting:
%%endtext

X: 2
T: Etude
T: second version
M: 4/4
L: 1/4
Q: "Adagio"
K: C
%
.C{DCB,}C.G{AGF}G|A>AG2|.F{GFE}F.E{FED}E|D>DC2|
%%sep 0.4cm 0.4cm 6cm
% the following line increases the character size
%%textfont * 20
%%center End of the example.
%%sep 0.4cm 0.4cm 6cm

```

## Typesetting example

This is an example of text inserted into an ABC file. This feature of abcm2ps allows for the writing of songbooks, music collections or other publications without having to resort to a word processor. Not bad, is it? Now let's write a brief musical example.

*Etude*

**Dolcemente**



Now we'll have a look at something more lively. To start with, let's switch fonts: from Bookman-Light to AvantGarde-Book. Here is the same Etude with a few small variations to make it more interesting:

*Etude*  
second version

**Adagio**



---

End of the example.

---

Hinweis: In gedruckten Noten wird meist Times Roman oder eine äquivalente Schriftart verwendet. Besser lesbar bei gleicher Schriftgröße ist jedoch Helvetica.

### 5.4 Unterbrechung des Notensystems

Um eine Besonderheit zu Beginn des Musikstückes (z.B. Originaltonart und Schlüssel) anzugeben oder eine Coda zu schreiben, kann das Notensystem mit dem %%staffbreak Kommando unterbrochen werden:



Wenn ein Notensystem aus mehreren Zeilen besteht, so muss das %%staffbreak Kommando in alle Notenzeilen eingefügt werden.

### 5.5 Mehrspaltiger Notensatz

Text und Musik können in mehreren Spalten auf einer Seite gesetzt werden. Die Kommandos %%multicol start, %%multicol new und %%multicol end definieren die Spalten.

`%%multicol start` sichert die momentanen Seitenränder und setzt die vertikale Position für den Spaltenbeginn. Jetzt können die Seitenränder geändert und das Material der ersten Spalte ausgegeben werden.

`%%multicol new` definiert die vertikale Position einer neuen Spalte und setzt die Ränder zurück. Die Ränder können nun wieder angepasst werden und die Musik ausgegeben werden. Diese Sequenz kann beliebig oft wiederholt werden.

Zum Schluss werden mit `%%multicol end` die Seitenränder wieder auf die ursprünglichen Werte vor `%%multicol start` zurückgesetzt und setzt die horizontale Position unter die ausgegebenen Spalten.

Dies klingt ziemlich kompliziert. Es jedoch einfacher als es klingt:

```
%%pagewidth 21cm
%%leftmargin 1cm
%%rightmargin 1cm
X: 1
L: 1/4
K: C
CDEF|GABc|cdef|gabc'|
%%multicol start
%%rightmargin 11cm
%%begintext justify
%%Sator arepo tenet opera rotas. Sator arepo tenet opera rotas.
%%Sator arepo tenet opera rotas. Sator arepo tenet opera rotas.
%%endtext
"^\left"CDEF|GABc|
%%text Left column (margins: 1, 11)
%%text Width: 21 - 1 - 11 = 9 cm
%%multicol new
%%leftmargin 13cm
%%rightmargin 2cm
%%begintext justify
%%Sator arepo tenet opera rotas. Sator arepo tenet opera rotas.
%%Sator arepo tenet opera rotas.
%%endtext
"^\right"cdef|gabc'|
%%text Right column (margins: 13, 2)
%%text Width: 21 - 13 - 2 = 6 cm
%%multicol end
CDEF|GABc|cdef|gabc'|
```



Sator arepo tenet opera rotas. Sator arepo tenet  
opera rotas. Sator arepo tenet opera rotas. Sator  
arepo tenet opera rotas.

Sator arepo tenet opera rotas.  
Sator arepo tenet opera rotas.  
Sator arepo tenet opera rotas.



Left column (margins: 1, 11)

Width:  $21 - 1 - 11 = 9$  cm



Right column (margins: 13, 2)

Width:  $21 - 13 - 2 = 6$  cm



## 5.6 Kopf- und Fußzeilen

Die folgenden Kommandos definieren den Text, der automatisch auf jeder Seite erscheinen soll:

`%%header` für die Kopfzeile und `%%footer` für die Fußzeile. Diese Kommandos, gefolgt von Text werden standardmäßig zentriert ausgegeben.

Es können drei Areale definiert werden: `left`, `centre` und `right`, mit unterschiedlichem Text in jedem Areal. Werden Areale definiert, sollte der Text in Anführungszeichen eingeschlossen werden.

Weiterhin können spezielle Symbole verwendet werden, um spezifische Informationen über das Stück auszugeben:

- `$D` gibt das aktuelle Datum und die Zeit aus.
- `$F` gibt den Dateinamen aus
- `$T` gibt den Titel des aktuellen Musikstückes aus
- `$P` gibt die Seitenzahl aus
- `$P0` und `$P1` gibt die Seitenzahl aus, jedoch nur wenn sie gerade bzw. ungerade ist.
- `$V` gibt `abcm2ps-` gefolgt von der Versionsnummer aus.
- `\n` zeigt den Start einer weiteren Textzeile an.

Die drei Felder müssen mit dem Tab-Zeichen voneinander getrennt werden. Wenn JEDABC benutzt wird, sei die Verwendung der Vorlagen unter `Mode/abcm2ps Options/page layout` empfohlen.

Hier ist ein Beispiel der Verwendung des Kommandos `%%footer` um die geraden Seitenzahlen auf der linken Seite, den Namen des Stückes in der Mitte und ungerade Seitenzahlen auf der rechten Seite auszugeben:

```
%%footer "$P0 $N $P1"
```

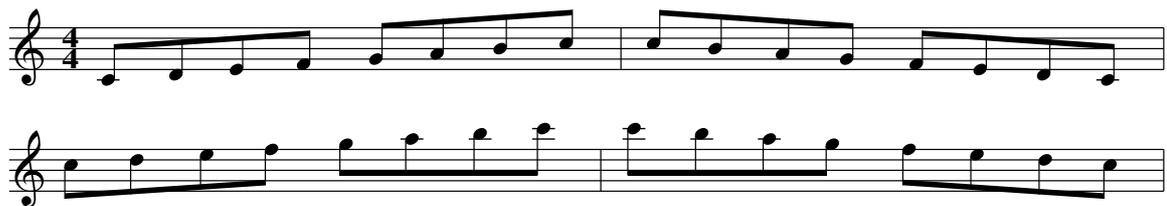
Zu beachten ist, dass die Felder nicht durch Leerzeichen, sondern durch Tabs voneinander getrennt sind!

## 5.7 Einfügen von Grafiken

Eine weitere interessante Option ist das Einfügen von EPS Grafiken, z.B. um ein Logo oder eine Zeichnung zum Notenblatt hinzuzufügen. Es wird das Kommando %%EPS, gefolgt von dem Namen der EPS-Datei, verwendet:

```
X: 1
T: Testing the use of my logo
K: C
CDEF GABc |cBAG FEDC |
cdef gabc'|c'bag fedc|
%%multicol start
%%leftmargin 1cm
%%rightmargin 10cm
%%text
%%text Beautiful music presented by...
%%multicol new
%%leftmargin 7cm
%%rightmargin 1cm
%%EPS logo.eps
%%multicol end
```

Testing the use of my logo



Beautiful music presented by...



Falls die Grafik nicht im EPS-Format vorliegt (z.B. JPG), muss sie zuvor ins EPS-Format konvertiert werden. Hinweise dazu finden sich in Kap. 8.2.

## 5.8 Format Dateien

Neben der Definition von Formatierungsparametern direkt im Quelltext ist es möglich, eine extra Datei zu erstellen, welche das zu verwendende Format beschreibt und von abcm2ps bei der Erzeugung des Notenblattes mit ausgewertet wird:

```
% format file
scale 0.8
topmargin 2 cm
titlefont Helvetica-Bold 13
subtitlefont Helvetica-Bold 10
% etc...
% end
```

Die Format Datei enthält die bereits beschriebenen Formatierungsparameter, jedoch ohne das vorangestellte Doppel %.

Um ein Musikstück auszugeben unter Benutzung einer Formatdatei `example.fmt`, welche sich im selben Verzeichnis wie der Quelltext befinden muss, wird die Option `-F` in der Kommandozeile von `abcm2ps` verwendet:

```
abcm2ps -O= -c -F example tune.abc
```

Die Benutzung einer Format Datei ist die beste Lösung, wenn eine Reihe von Musikstücken mit gleichem Seitenlayout und Stil erzeugt werden soll.

## 5.9 Numerierung von Takten und Seiten

Normalerweise wird auf einem Notenblatt nur der erste Takt jeder Zeile nummeriert. Das kann durch Verwendung des Kommandos `%%measurenb 0` erreicht werden. Um alle Takte zu nummerieren, wird `%%measurenb 1` verwendet. Soll alle  $n$  Takte eine Takt Nummer eingefügt werden, so wird `%%measurenb n` verwendet. Takt Nummern beginnen mit 1, es sei denn, der erste Takt ist unvollständig. In diesem Fall zählt der Auftakt als Takt 0.

Die Seitennummerierung wird mit der Kommandozeilenoption `-N` von `abcm2ps` gesteuert. Mögliche Werte sind:

- 0: Seitennummerierung abgeschaltet
- 1: Seitenzahlen oben links.
- 2: Seitenzahlen rechts.
- 3: Seitenzahlen links auf geraden Seiten und rechts auf ungeraden Seiten
- 4: Seitenzahlen rechts auf geraden Seiten und links auf ungeraden Seiten

## 5.10 Platz sparen

Ein häufig auftretendes Problem ist der Ausdruck eines Musikstückes auf der kleinst möglichen Zahl von Seiten. Wenn das Seitenlayout und die Ränder festgelegt sind, gibt es folgende Parameter um weiteren Platz zu sparen:

- Zunächst kann mit der Anweisung `%%scale <Faktor>` die Ausgabegröße aller Elemente zu ändern. Der Faktor ist initial auf 0,7 eingestellt. Ein größerer Faktor vergrößert die Ausgabe, ein kleinerer verkleinert die Ausgabe.
- Der Platz zwischen den Notenzeilen kann mit `%%staffsep` und `%%sysstaffsep` reduziert werden. Weiterhin kann mit anderen Kommandos der vertikale Abstand für Titel, Untertitel, Text usw. angepasst werden.
- Wird die `-c` Option (Kommandozeilenparameter) benutzt, kann die Anweisung `%%maxshrink <Faktor>` benutzt werden, um den horizontalen Abstand zwischen den Noten zu beeinflussen. Die Kompression ist maximal mit `Faktor=1` und minimal mit `Faktor=0`;
- Um Bögen flacher zu zeichnen kann das Kommando `%%slurheight` mit Werten kleiner 1 benutzt werden.
- In einigen Fällen kann die Kombination aus `%%notespacingfactor` und `%%maxshrink` gute Ergebnisse produzieren. Normalerweise ist der Notenabstand proportional zur Notendauer. `%%notespacingfactor 1` bewirkt, dass alle Noten den gleichen Abstand haben.
- In einer Datei mit mehreren Stücken kann `%%topspace 0` benutzt werden.

Bei der Anwendung der dargestellten Verfahren sollte beachtet werden, dass nicht jeder Adleraugen besitzt: Der Ausdruck eines Stückes mit einer kleinen Skalierung macht es für den Musiker nicht leichter.

## 5.11 Anpassungen durch fortgeschrittene Nutzer

Achtung: dieses Kapitel ist für versierte Computernutzer gedacht!

Abcm2ps besitzt eine sehr mächtige Eigenschaft: Es kann Postscript Routinen definieren und/oder modifizieren. Dafür gibt es eine Reihe von Kommandos, welche neue Symbole oder Routinen im Quelltext definieren. In abcm2ps definierte Postscript-Routinen können benutzt und neue Routinen hinzugefügt werden.

Es ist klar, dass diese Möglichkeiten nur von Programmierern umfassend nutzbar sind. Es kann notwendig und nützlich sein, sich mit den Quelltexten von abcm2ps vertraut zu machen und sich den Postscript Code anzuschauen, der von abcm2ps erzeugt wird.

### 5.11.1 Neue Postscript Routinen

Das `%%postscript` Kommando, gefolgt von Postscript code fügt neue Routinen hinzu oder definiert existierende Routinen neu.

Als Beispiel definieren die folgenden Kommandos die Routine `dlw` neu, so dass alle Linien auf dem Notenblatt dünner ausgegeben werden:

```
%%postscript /dlw
%%postscript {0.2 setlinewidth} bdef % default: 0.7
```

Die Postscript-Routinen von abcm2ps sind in der Quelldatei `syms.c` definiert.

### 5.11.2 Begleitakkorde in italienischer Notation

Eine gute Anwendung für `%%postscript` ist Neudefinition der Routine welche Begleitakkorde ausgibt um sie mit italienischen Noten auszugeben.

Die Beispieldatei `deco.abc`, welche in der abcm2ps Distribution enthalten ist, verwendet folgenden Code:

```
% -- latin guitar chords
%%postscript /gcshow{
%%postscript   -5 0 RM
%%postscript   dup 0 get
%%postscript   dup 65 eq {(La) show}
%%postscript   {dup 66 eq {(Si) show}
%%postscript   {dup 67 eq {(Do) show}
%%postscript   {dup 68 eq {(Re) show}
%%postscript   {dup 69 eq {(Mi) show}
%%postscript   {dup 70 eq {(Fa) show}
%%postscript   {dup 71 eq {(Sol) show}
%%postscript   {tempstr 0 2 index put tempstr show
%%postscript   } ifelse
%%postscript   } ifelse
%%postscript   } ifelse
```

```

%%postscript      } ifelse
%%postscript      } ifelse
%%postscript      } ifelse
%%postscript      } ifelse
%%postscript      pop dup dup length 1 sub 1 exch getinterval
%%postscript      {dup 129 eq {sharp_glyph}
%%postscript      {dup 130 eq {flat_glyph}
%%postscript      {dup 131 eq {nat_glyph}
%%postscript      {tempstr 0 2 index put tempstr show}
%%postscript      ifelse}
%%postscript      ifelse}
%%postscript      ifelse pop}
%%postscript      forall
%%postscript      pop}!

```



### 5.11.3 Neue Symbole

Das `%%deco` Kommando erzeugt neue Ausdruckssymbole mit Hilfe der in `abcm2ps` definierten Postscript-Routinen oder selbstgeschriebener Routinen.

Die Syntax ist wie folgt:

```
%%deco <name> <type> <ps> <h> <wl> <wr> <string>
```

wobei

- `<name>` der Name des neuen Symbols ist (ohne Klammern)
- `<type>` ist ein Integer Wert der den Symboltyp angibt. Werte zwischen 0 und 2 kennzeichnen ein Symbol nahe der Note innerhalb des Notensystems, Werte zwischen 3 und 5 kennzeichnen ein Symbol außerhalb des Systems. Die Werte 6 und 7 kennzeichnen Ausdruckssymbole verbunden mit dem Notensystem.

Um eine Vorstellung von der Positionierung von Symbolen zu erhalten, sind nachfolgend einige vordefinierte Symbole mit ihren `<type>`-Werten aufgelistet:

- 0: `!tenuto!` oder der staccato Punkt
- 1: `!slide!`;
- 2: `!arpeggio!`;
- 3, 4: Generische Ausdrücke
- 5: `!trill(! oder !trill)!`;
- 6: generisch
- 7: lange Dynamik Symbole

- `<ps>` ist der Name der Postscript-Routine welche das Symbol zeichnet. Es kann eine durch den Nutzer definierte Routine oder eine von `abcm2ps` zur Verfügung gestellte Routine sein.
- `<h>` Symbolhöhe in Punkten
- `<wl>` und `<wr>` werden nicht benutzt
- `<string>` ist ein optionaler Text

Im folgenden wird nun ein Beispiel aus der Datei `deco.abc` angeführt. Es werden einige neue Dynamiksymbole unter Verwendung vordefinierter Routinen definiert und verwendet:

```

%%deco fp 6 pf 20 0 0 fp
%%deco mp 6 pf 20 0 0 mp
%%deco (f) 6 pf 20 0 0 (f)
%%deco (ff) 6 pf 20 0 0 (ff)
X: 1
T: New dynamics symbols
K: C
!fp!CDEF GABc|!mp!CDEF !(f)!GABc|!(ff)!CDEF !ff!GABc|

```

### New dynamics symbols



Die %%deco Zeilen definieren vier neue Symbole: !fp!, !mp!, !(f)! and !(ff)!.

Der folgende Quelltext definiert drei neue Symbole: ein Notenbezogenes und zwei Systembezogene Symbole, über und unter dem System. Das erste Symbol !tu! ist ein dreieckiges Staccato-Symbol. Es verwendet die neue Routine newdot. Die anderen Symbole !rtoe! und !ltoe! nutzen die Routine toe und definieren Symbole ähnlich einem ^ über und unter dem System.

```

%%postscript /newdot { % usage: x y newdot
%%postscript M 1.2 2.5 rmoveto -2.4 0 rlineto
%%postscript 1.2 -5 rlineto fill } bdef
%%deco tu 0 newdot 5 0 0
%%postscript /toe { % usage: x y toe
%%postscript M 5 0 rmoveto
%%postscript -5 5 rlineto -5 -5 rlineto currentpoint stroke
%%postscript } bdef
%%deco rtoe 6 toe 5 0 0
%%deco ltoe 3 toe 5 0 0
X: 1
K: C
!tu!C!tu!D!tu!E!tu!F GABc|!ltoe!c'!rtoe!bag .f.e.d.c|!ltoe!C4 z4|

```



## 5.11.4 Hinzufügen neuer Schriftarten

Standard GhostScript Schriftarten reichen normalerweise für die meisten Nutzer. Falls jedoch eine spezielle Schriftart verwendet werden soll können diese verwendet werden. Es können jedoch nur Postscript Schriftarten verwendet werden. Für Details siehe Anhang D3

Eine sehr gute Web-Seite mit einer großen Anzahl freier Pstscriptschriftarten ist

<http://www.apostrophiclab.com>

Im folgenden wird gezeigt, wie eine neue Schriftart "Labrit" verwendet wird. Ausgangspunkt ist das heruntergeladene Archiv labrit1.zip.

Zunächst wird das Archiv entpackt und die Datei LABRIT.pfb in das Verzeichnis kopiert, welches die Ghostscript Schriftarten enthält. (Weitere mögliche Dateitypen sind .gsf und .pfa). Normalerweise ist das das Verzeichnis C:\gs\gs8.00\font auf Windows Systemen (Die Nummer entspricht der installierte Ghostscriptversion). Auf Linux und anderen Unix-Systemen sind die Schriften unter /usr/share/fonts/default/ghostscript/ abgelegt.

Jetzt muß die Schriftartenliste von GhostScript editiert werden. Unter Windows ist das die Datei C:\gs\gs8.00\lib\Fontmap.GS, unter Linux ist es /usr/share/ghostscript/6.52/lib/Fontmap.GS. Am Ende der Datei ist folgender Eintrag hinzuzufügen:

```
/Labrit (LABRIT__.pfb) ;
```

Dadurch wird eine neue Schriftart Labrit definiert. Es kann auch ein Aliasname, d.h. ein alternativer Name für die gleiche Schriftart definiert werden:

```
/Fraktur /Labrit ;
```

Jetzt kann die neue Schriftart in ABCPLUS Dateien verwendet werden. Wie bereits beschrieben ist zunächst eine Deklaration der Schriftart mit der Anweisung %%font nötig:

```
%%font Labrit
%%titlefont Labrit 24
%%textfont Labrit 18
%%composerfont Fraktur 16 % alias
%%vocalfont Fraktur 12
X: 1
T: Test: Labrit font (Fraktur)
L: 1/4
K: C
%
CDEF|GABc|cBAG|FEDC|
w: Do Re Mi Fa... ||||
%%text ABCDEFGHIJKLMNOPQRSTUVWXYZ
%%text abcdefghijklmnopqrstuvwxyz 1234567890
```

### Test: Labrit font (Fraktur)



ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz 1234567890

Viele Schriftarten, die aus dem Internet heruntergeladen werden können, sind nicht komplett (enthalten z.B. nur Großbuchstaben oder besitzen keine deutschen Umlaute). Es können auch nicht alle Schriftarten frei verwendet werden und die Qualität der Schriften ist sehr unterschiedlich. Ein nützliches Programm zur Konvertierung von True Type Schriftarten in Postscript Schriftarten ist ttf2pt1, verfügbar über folgende Internetadresse:

<http://ttf2pt1.sourceforge.net/>

## 5.12 Flöten Fingersätze (tin whistle)

Die Ursprünge der ABC Musiknotation sind eng mit der Folk-Musik verbunden.

Die Flöte, die Gegenstand dieses Kapitels ist, ist eine kleine Metallflöte mit sechs Löchern, die häufig in der traditionellen angelsächsischen Musik verwendet wird. Sie ist preiswert, einfach zu erlernen und es macht Spaß, darauf zu spielen. Der Kauf einer solchen Flöte sei wärmsten empfohlen!

Die Option

-W <Stimme><Tonart>

auf der Kommandozeile von abcm2ps gibt die Fingersätze, die zu <Stimme> gehören, unter dem Notensystem der Stimme aus. Bei einstimmigen Stücken ist Stimme 1. <Tonart> gibt die Tonlage der Flöte an. Der Grundton der Flöte wird als Großbuchstabe angegeben, ein eventuelles Vorzeichen als Kleinbuchstabe danach. Auch hier ist wieder zu beachten, dass der Ton H im englischen Sprachraum mit B bezeichnet wird. Für eine B-Flöte wird Bb als Tonart angegeben. Zwischen Stimme und Tonart dürfen keine Leerzeichen eingefügt werden.

Als Beispiel soll eine einfache Tonleiter gezeigt werden. Mit der Option -W 1Bb ergibt sich folgende Ausgabe:

### Whistle

The image shows a musical score for a tin whistle in B-flat major, 4/4 time. The notation consists of a single staff with a treble clef and a 4/4 time signature. The melody is a simple scale: C4, D4, E4, F4, G4, A4, Bb4, C5, Bb4, A4, G4, F4, E4, D4, C4. Below the staff, there are 16 vertical columns of circles representing fingerings for each note. The first column is labeled 'WHISTLE Bb' and has a downward arrow above it. The circles are arranged in a 2x8 grid. The top row of circles represents the left hand (fingers 1-4) and the bottom row represents the right hand (fingers 1-4). A solid black circle indicates a finger to be pressed down, while an open circle indicates a finger to be held up. The fingering for the first note (C4) is: left hand (1, 2, 3, 4) all down, right hand (1, 2, 3, 4) all up. The fingering for the final note (C4) is: left hand (1, 2, 3, 4) all down, right hand (1, 2, 3, 4) all up with a '+' sign below each circle.

## 6 Teil V abcMIDI

### 6.1 Erzeugen von MIDI-Daten

Eine MIDI-Datei ist grob gesagt so etwas wie ein elektronisches Notenblatt. Es enthält die Informationen die benötigt werden, um MIDI Instrumenten (oder einem Software Player) mitteilen zu können welche Noten wann und wie gespielt werden sollen.

Im Gegensatz zum menschlichen Interpreten benötigen elektronische Instrumente und Computer exakte Informationen, wie sie spielen sollen. Normale Notenblätter sind oft nicht ganz eindeutig, z.B. ist bei der Angabe eines Trillers nicht festgelegt, wie viele kurze Noten gespielt werden sollen.

Weiterhin ist zu beachten, dass ein Notenblatt im PDF-Format auf allen Computern gleich aussieht. Die Ausgabe der Interpretation einer MIDI-Datei ist stark abhängig von der Soundkarte des Computers oder des angeschlossenen Instruments.

Die Erzeugung von MIDI-Daten aus dem ABC PLUS Quelltext ist eine gute Möglichkeit, sofort einen Eindruck von der geschriebenen Musik zu erhalten. Zu diesem Zweck wird das Programm `abc2midi` benutzt. Es ist wie `abcm2ps` ebenfalls ein Programm, welches auf der Kommandozeile ausgeführt wird und einen ABC-Quelltext in eine MIDI-Datei konvertiert, die dann mit einem auf dem Computer vorhandenen MIDI Player abgespielt werden kann. `abc2midi` erzeugt so viele MIDI-Dateien wie Stücke in einem abc-Quelltext vorhanden sind. Dabei wird die X:-Feld Indexnummer zum Dateinamen der MIDI-Datei hinzugefügt:  
`file1.mid, file2.mid,.....`

`abc2midi` ist in JEDABC integriert. Es ist nur eins der Programme, welche zum abcMIDI-Paket gehören. Neben `abc2midi` gehören folgende Programme dazu:

- `abc2abc`: Überprüfung, Formatierung und Transposition von ABC-Quelltexten.
- `midi2abc`: Konvertierung von MIDI-Dateien in ABC-Quelltext
- `yaps`: Ein Formatierung ähnlich `abcm2ps`, jedoch weniger leistungsfähig.

`abc2midi` benutzt den Meta-Kommentar `%%MIDI` mit verschiedenen Parametern zur Definition der MIDI-Eigenschaften.

Hinweis für Linux Benutzer: Wenn die Unterstützung der jeweils vorhandenen Soundcard durch Linux Probleme bereitet (oft der Fall bei billigen Karten) kann versucht werden, mit dem MIDI player `Timidity++` (<http://www.onicos.com/staff/iz/timidity/>). Dieser Player benutzt nicht den Hardware Sequenzer der Soundkarte.

#### 6.1.1 Stimmen und Instrumente

Als Beispiel in diesem Kapitel soll das bereits in Kapitel 4.2 dargestellte `Ave Verum` dienen. Bei der Konvertierung mit `abc2midi` wird eine MIDI-Datei erzeugt, welche für alle Stimmen das MIDI-Instrument Nummer 1 verwendet: Klavier. In vielen Fällen wird das ausreichen. `abc2midi` kann jedoch viel mehr.

Eines der wichtigsten `abc2midi`-Befehle ist `%%MIDI program` welcher eine Stimme mit einem bestimmten Instrument verbindet. Im folgenden ist eine Version des `Ave Verum` mit den Erweiterungen für `abc2midi` dargestellt:

```

X: 1
T: Ave Verum
C: W. A. Mozart
M: 4/4
L: 1/4
Q: "Adagio"
%%staves [(S A) (T B)] {(MD1 MD2) (MS1 MS2)}
V: S clef=treble name="Soprano" sname="S"
V: A clef=treble name="Alto" sname="A"
V: T clef=bass name="Tenor" sname="T"
V: B clef=bass name="Bass" sname="B"
V: MD1 clef=treble name="Organ"
V: MD2 clef=treble
V: MS1 clef=bass
V: MS2 clef=bass
K: D
%
%%MIDI program 1 53 % Choir Oohs
%%MIDI program 2 53
%%MIDI program 3 53
%%MIDI program 4 53
%%MIDI program 5 19 % Church Organ
%%MIDI program 6 19
%%MIDI program 7 19
%%MIDI program 8 19
... body of transcription ...

```

Den acht Stimmen S A T B MD1 MD2 MS1 MS2 werden von abc2midi automatisch die Kanalnummern 1 bis 8 zugewiesen. Die %%MIDI program Befehle verbinden dann jede Stimme mit einem geeigneten Instrument, MIDI 54 ("Choir Oohs") oder 20 ("Church Organ").

Die Nummern, die zur Angabe eines Instrumentes verwendet werden, sind abhängig von der MIDI-Implementation des MIDI-Players oder Instruments. Für den „General MIDI“-Standard, der heute in fast allen Keyboards und Tongeneratoren verwendet wird, ist die Zuordnung zwischen Instrument und Programmnummer in Anhang F aufgelistet. Zu beachten ist, dass der GM-Standard Programmnummern zwischen 1 und 128 vorsieht. abc2midi erwarten Nummern zwischen 0 und 127. Es ist also jeweils 1 abzuziehen von den im Anhang gelisteten Programmnummern. Wird kein Instrument angegeben, so wird den Stimmen automatisch Instrument 1 zugewiesen.

Beim Abspielen der MIDI-Datei wird man jetzt feststellen, dass irgendetwas nicht stimmt. Der Grund dafür ist Thema des nächsten Kapitels.

### 6.1.2 Der Bass Schlüssel

Der größte Unterschied zwischen abcm2ps und abc2midi ist die Art und Weise, wie unterschiedliche Notenschlüssel interpretiert werden. In Kapitel 3.1 wurde gezeigt dass abcm2ps Notenzeilen mit dem Bassschlüssel versieht, wenn K:bass angegeben wurde. Die Noten werden dann einfach zwei Notenlinien tiefer im System ausgegeben.

Der Ansatz von abc2midi ist komplett anders:

Die Tohhöhe der Noten bleibt immer gleich, unabhängig vom Schlüssel. Das bedeutet, wenn ein Stück mit Bass-Schlüssel geschrieben wurde und die Noten ohne Kommas, dann werden die Noten von abc2midi zwei Oktaven zu hoch ausgegeben.

Zum Glück gibt es eine einfache Lösung für dieses Problem:

Zum K: oder V:-Feld muss die Definition octave=-2 hinzugefügt werden. Das folgende Beispiel wird korrekt ausgegeben und korrekt wiedergegeben:

```
X: 1
L: 1/4
K: C bass octave=-2
cdef|gabc' |
```

### 6.1.3 Begleitakkorde

Die Begleitakkorde welche bereits in Kapitel 3.6 beschrieben wurden, werden von abc2midi benutzt, um eine Begleitung zur Melodie zu generieren. Momentan werden folgende Akkorde erkannt:

```
m 7 m7 maj7 M7 6 m6 aug + aug7 dim dim7 9
m9 maj9 M9 11 dim9 sus sus9 7sus4 7sus9 5
```

Zusätzliche Akkorde können mit %%MIDI chordname definiert werden. Dies wird später noch näher erläutert.

In Abhängigkeit von der Taktart werden Begleitungen als Folge von fcz für jeden Takt generiert. „f“ steht für die Grundnote (engl. fundamental) , „c“ für den Akkord (engl. chord) und z für Pause. Z.B. für einen 4/4-Takt wird folgende Begleitung generiert: fczfzcz. Zu beachten ist, dass die fcz-Folge nicht von der Anzahl der Viertelnoten in einem Takt abhängt. Auch der Wert des L:-Feldes spielt keine Rolle. Die Sequenzen werden so interpretiert, dass sie einen Takt ergeben. Die Folgen fcz, f2c2z2 und f4c4z4 führen daher zum gleichen Ergebnis.

Im folgenden ist ein Stück mit Begleitakkorden im Quelltext und der Ausgabe durch abc2midi dargestellt:

```
X: 1
M: 4/4
L: 1/4
K: C
%
"C"CDEF | "G"GABc | "C"C2"G"E2 | "C"Czz2 |
```

Die fcz-Folge kann mit dem %%MIDI gchord Kommando modifiziert werden wenn nötig.  
Dies ist das gleiche Stück mit einer einfacheren Begleitung:

```
X: 1
M: 4/4
L: 1/4
K: C
%%MIDI gchord c4c4
%
"C"CDEF| "G"GABc| "C"C2"G"E2| "C"Cz z2|
```

Die Begleitfolge wurde so geändert, dass pro Takt nur zwei Akkorde ausgegeben werden.  
Für die üblichen Takarten werden automatisch folgende Sequenzen verwendet:

```
3/4 fzczcz
4/4 fzc zfcz
6/8 fzc fzc
9/8 fzc fzc fzc
```

Um die Instrumente und Klangeigenschaften für die Begleitung zu ändern, können folgende Befehle verwendet werden:

```
%%MIDI chordprog   Instrument für den Akkord
%%MIDI bassprog    Instrument für den Grundton
%%MIDI chordvol    Laustärke für den Akkord
%%MIDI bassvol     Lautstärke für den Grundton
```

Um die Begleitung während des Stückes aus- und wieder einzuschalten können die Befehle %%MIDI gchordoff und %%MIDI gchordon verwendet werden.

Die Akkorde werden auch weitergespielt wenn die Melodie aufhört. Das folgende Stück hat keine Melodie sondern nur Begleitakkorde:

```
X: 1
T: La Folia
M: 3/4
L: 1/4
Q: 80
K: Dm
%%MIDI gchord ccz
%%MIDI chordprog 24 % guitar
"Dm"z3| "A"z3| "Dm"z3| "C"z3| "F"z3| "C"z3| "Dm"z3|\
%%MIDI gchord c3
"A"z3|
%%MIDI gchord czc
"Dm"z3| "A"z3| "Dm"z3| "C"z3| "F"z3| "C"z3| "A"z3|\
%%MIDI gchord c3
"Dm"z3|]
```

Nun soll ein Stück betrachtet werden, welches zwar im 4/4-Takt notiert ist, jedoch mit einem völlig abweichenden Rhythmus: das berühmte brasilianische Stück „The girl from Ipanema“ von Antonio Carlos Jobim:

```

X: 1
T: Garota De Ipanema
T: (The Girl From Ipanema)
C: Antonio Carlos Jobim
M: 4/4
L: 1/8
K: F
P:A
"Fmaj7" !p!G2 GE E2 ED|G2 GE EE DG-|"G7"G2 GE EE DG-|
G2 GE EE DF-|"Gm7"F2 FD DD CE-|"Gb7"E2 EC CC B,C-|
[1"Fmaj7"C8|"Gb7"z8 :|[2"Fmaj7"C8 |z8||
P:B
"Gb7"F8-|(3F2_G2F2 (3:2:3_E2F2E2|"B7"_D3 _E-E4-|_E6 z ^G-|
"F#m7"^G8-|(3^G2A2G2 (3^F2G2F2|"D7"E3 ^F-F4-|^F6 z A-|
"Gm7"A8-|(3A2B2A2 (3:2:3G2A2G2|"Eb7"F3 G-G4-|G4 (3z2A2B2|
"Am7"(3c2C2D2 (3E2F2G2|"D7"^G3 A3 z2|"Gm7" (3B2B,2C2
(3D2E2F2|"C7" ^F3 G3 z2 ||
P:C
"Fmaj7"G3 E EE DG-|G2 GE- EE DG-|"G7"G2 GE EE DG-|G2 GE EE DA-|
"Gm7"A2 AF FF Dc-|"Gb7" c2 cE (3E2E2D2|"Fmaj7" E8-|E2 z6|
P:D
z8||

```

Wenn dieses Stück so wie geschrieben von abc2midi konvertiert wird, hat es wenig mit dem Original zu tun. Eine Bossa Nova besitzt einen völlig anderen Rhythmus. Es wird eine andere fcz-Folge benötigt. Folgende Zeilen sind nach dem K:-Feld einzufügen:

```

%%MIDI program 67 % Baritone Sax
%%MIDI gchord fzcffczc % bossa nova (approximate)
%%MIDI chordvol 80
%%MIDI bassvol 80
%%MIDI chordprog 25 % Steel String Guitar
%%MIDI bassprog 25

```

und folgendes Kommando unmittelbar nach dem P:D-Feld:

```

%%MIDI gchord c2

```

Mit diesen Änderungen klingt das Ergebnis schon eher nach Bossa Nova.

## 6.1.4 Neue Begleitakkorde

Der `%%MIDI chordname` Befehl erlaubt, die Noten eines Begleitakkordes zu ändern oder neue Akkorde zu definieren.

```
%%MIDI chordname <chord name> <n1> <n2> <n3> <n4> <n5> <n6>
```

wobei

<code>&lt;chord name&gt;</code>	der Name des Akkordes ist,
<code>&lt;n1&gt;</code>	Der Grundton ist
<code>&lt;n2&gt;...&lt;n6&gt;</code>	die anderen Noten in Halbtonschritten relativ zum Grundton sind

Die folgenden Zeilen definieren die Akkorde "4" und "5+":

```
%%MIDI chordname 4 0 5 7 12
%%MIDI chordname 5+ 0 4 8 12
```

Jetzt können die neuen Akkorde in Begleitungen verwendet werden: "C4", "G5+" usw.

## 6.1.5 Schlagzeug

Zusätzlich zu Begleitakkorden können auch Schlaginstrumente über den Befehl `%%MIDI drum` gesteuert werden. Die Syntax dieses Befehles ist ähnlich dem Befehl `%%MIDI gchord`. Dem `%%MIDI drum` – Befehl wird eine dz-Folge angefügt, welche jeweils einen Schlag(d) und eine Pause(z) darstellt. Nach der dz-Folge werden die Nummern der Percussionsinstrumente für jeden Schlag und dessen Lautstärken mit den Werten 0..127 definiert.

Die Schlagzeugbegleitung wird mit `%%MIDI drumon` eingeschaltet und mit `%%MIDI drumoff` aus.

Das folgende Lied wird von großer Trommel und Hi-Hat begleitet:

```
X: 1
M: 4/4
L: 1/4
K: C
% sequence instrument volume
%%MIDI drum dddd 36 46 36 46 80 100 80 100
% bass drum 1, open hi-hat
%%MIDI drumon
CDEF|GABc|\
%%MIDI drumoff
cdef|\
%%MIDI drumon
gabc'|
```

Es ist nicht möglich gebrochene Längen zu definieren: um einen Rhythmus wie z.B. (3ddd d/d/d/d/ zu erzeugen, muss dieser als d4d4d4d3d3d3d3 definiert werden.

Nun ein etwas komplexeres Beispiel (riff.mid):

```
X: 1
M: 4/4
```

```

T: Riff
%%MIDI program 1 25 % Steel String Guitar
Q: 1/4=160
K: C
%%MIDI drum dzddd2dz 35 39 39 35 39 127 80 80 127 80
% Bass Drum 1 + Electric Snare
%%MIDI gchord ccccccc
%%MIDI drumon
"C"CC EE GG AA|_BB AA GG EE|
CC EE GG AA|_BB AA GG EE|
"F"FF AA cc dd|"F7"_ee dd cc AA|
"C"CC EE GG AA|_BB AA GG EE|
%%MIDI gchordoff % no chords
"G"GG BB dd Bd|"F7"FF AA cc Ac|
%%MIDI gchordon % turn chords back on
"C"CC EE GG AA|_BB AA GG EE|
%%MIDI gchord c8
%%MIDI program 1 60 % Brass Section
%%MIDI drumoff
!fermata!"C"[C8E8G8c8]|

```

## 6.1.6 Schlaginstrumente

MIDI Dateien enthalten Daten für verschiedene MIDI-Kanäle. Für jeden Kanal können wiederum mehrere Spuren (engl. Tracks) definiert sein. In allen MIDI-Kanälen außer dem Kanal 10 gibt es pro Kanal ein Instrument, dem alle Noten aus allen diesem Kanal zugehörigen Spuren zugeordnet sind. Bei Kanal 10 ist dies nun anders: Jede Note ist einem anderen Schlaginstrument zugeordnet.

Der Instrumentenliste aus Anhang G folgend hat das folgende Stück eine Melodie, gespielt mit Klavier, und eine Begleitung aus TomTom, HiHat und Triangel (accomp.mid):

```

X: 1
L: 1/4
Q: 1/4 = 120
V: 1 name="Piano"
V: 2 clef=perc name="Low Tom"
V: 3 clef=perc name="Open Hi Hat"
V: 4 clef=perc name="Open Triangle"
K: C
%
[V: 1] CDEF |GABc | % Piano
[V: 2] \
%%MIDI channel 10
A,,zA,,z |A,,zA,,z | % Low Tom
[V: 3] \
%%MIDI channel 10
z^A,,z^A,, |z^A,,z^A,, | % Open Hi Hat
[V: 4] \
%%MIDI channel 10
a/a//a//a/a/a/a/a/a/a/a/|a/a//a//a/a/a/a/a/a/a/| % Open Triangle

```

Zu beachten ist, dass der Befehl %%MIDI channel 10 nach jedem V:-Feld, das eine neue Stimme beginnt, eingefügt werden muss.

Wenn wir versuchen, dieses Stück als Notenblatt auszugeben, erhalten wir leider nicht ganz das erwartete Ergebnis:

The image shows a musical score for four percussion instruments: Piano, Low Tom, Open Hi Hat, and Open Triangle. The score is in 4/4 time and consists of two measures. The Piano part is written on a treble clef staff with a melody of quarter notes. The Low Tom part is written on a bass clef staff with a bass line of quarter notes. The Open Hi Hat part is written on a bass clef staff with a pattern of eighth notes and 'x' marks. The Open Triangle part is written on a bass clef staff with a pattern of eighth notes and 'x' marks.

Leider besitzt `abc2ps` nur begrenzte Unterstützung für die Notation von Perkussionsstimmen. Zukünftige Versionen werden volle Unterstützung für Perkussionsstimmen beinhalten.

## 6.2 Erweiterte Nutzung von P:

Die ersten zwei Takte des folgenden Stückes werden dreimal wiederholt (angegeben mit `l::` `::l`), danach folgen die letzten beiden Takte:

```
X: 1
T: Repeats
T: printed only
L: 1/4
K: C
|:: CDEF|GABc ::|C2c2|Czz2|
```

Die generierte MIDI-Datei enthält nur eine Wiederholung, `abc2midi` interpretiert die doppelten Wiederholungszeichen nicht korrekt. Das ist jedoch nur ein Beispiel wo die korrekte Notation des gedruckten Notenblattes nicht mit der Übersetzung durch `abc2midi` übereinstimmt. In diesen Fällen kann mit dem P:-Feld (Kap. 3.5) gearbeitet werden. Das P:-Feld wird nicht nur benutzt um Teile (engl. Parts) zu definieren sondern auch anzugeben, in welcher Reihenfolge die Teile gespielt werden sollen. Das obige Beispiel kann folgendermaßen umgeschrieben werden:

```
X: 1
T: Repeats
T: printed and played
L: 1/4
P: A3.B % <- play part A 3 times, then B
K: C
|:: [P:A]CDEF|GABc ::|[P:B]C2c2|Czz2|
```

Ein Nachteil dieser Form des Quelltextes ist, dass die Definition der Teile auch im gedruckten Notenblatt erscheint. Ist das nicht erwünscht, kann der Befehl `%%printparts 0` benutzt werden.

## 6.2.1 Wiederholungen

Kapitel 6 begann mit der Aussage, dass MIDI-Dateien detaillierter als Notenblätter sein müssen. Das folgende Stück demonstriert dies nochmals:

```
X: 1
T: Manfrina di Camposilvano
M: 6/8
L: 1/8
Q: 1/4 = 160
K: G
DEF|: G2z DEF|G2z DEF|G2DG2D|G2z DEF|!segno!G2BA2c|B2d dcB|
ABc cBA|B2G DEF|G2BA2c|B2d dcB |ABc cBA|1 G3 DEF:|2 G3z Bc|
|: dz B d2B|e2ce2c|f2e d2 f|gdBz Bc|d zB d2B|e2c e2c|
f2e d2f|1 g3zBc|2 g3 DEF !D.S.!:|g3 z3|]
```

### Manfrina di Camposilvano

Scheinbar kein Problem. Aber wenn der Quelltext in eine MIDI-Datei konvertiert wird, stimmt die generierte Musik nicht mit dem Notenblatt überein. Das Problem ist, dass abc2midi nicht in der Lage ist zu erkennen, was !segno! und !D.S.! bedeuten. Erschwerend kommt hinzu, dass !segno! sich in einem wiederholten Teil, aber nicht zu Beginn, befindet. Um eine korrekte Wiedergabe zu erhalten müssen die logischen Teile des Stückes bestimmt werden und mit P:-Anweisungen deklariert werden.

Eine korrekte Variante ist folgende:

```
%%printparts 0
X: 1
T: Manfrina di Camposilvano
M: 6/8
L: 1/8
Q: 1/4 = 160
P: ABCD.BCE.FGFH.CDCE.FGFI
K: G
[P:A] DEF|:[P:B] G2z DEF|G2z DEF|G2DG2D|G2z DEF|
```

```
[P:C] !segno!G2BA2c|B2d dcB|ABc cBA|B2G DEF|G2BA2c|
B2d dcB |ABc cBA|1 [P:D] G3 DEF:|2 [P:E] G3z Bc|
|:[P:F] dz B d2B|e2ce2c|f2e d2 f|gdBz Bc|d zB d2B|
e2c e2c|f2e d2f|1 [P:G] g3zBc|2 [P:H] g3 DEF !D.S.!:|
[P:I] g3 z3|]
```

### 6.3 *midi2abc*

Dieses Programm konvertiert eine MIDI-Datei in einen ABC PLUS Quelltext. Im Prinzip funktioniert dies, jedoch ist meist mehr oder weniger Nacharbeit erforderlich, um das Stimmen-Layout anzupassen und Formatierungsparameter hinzuzufügen.

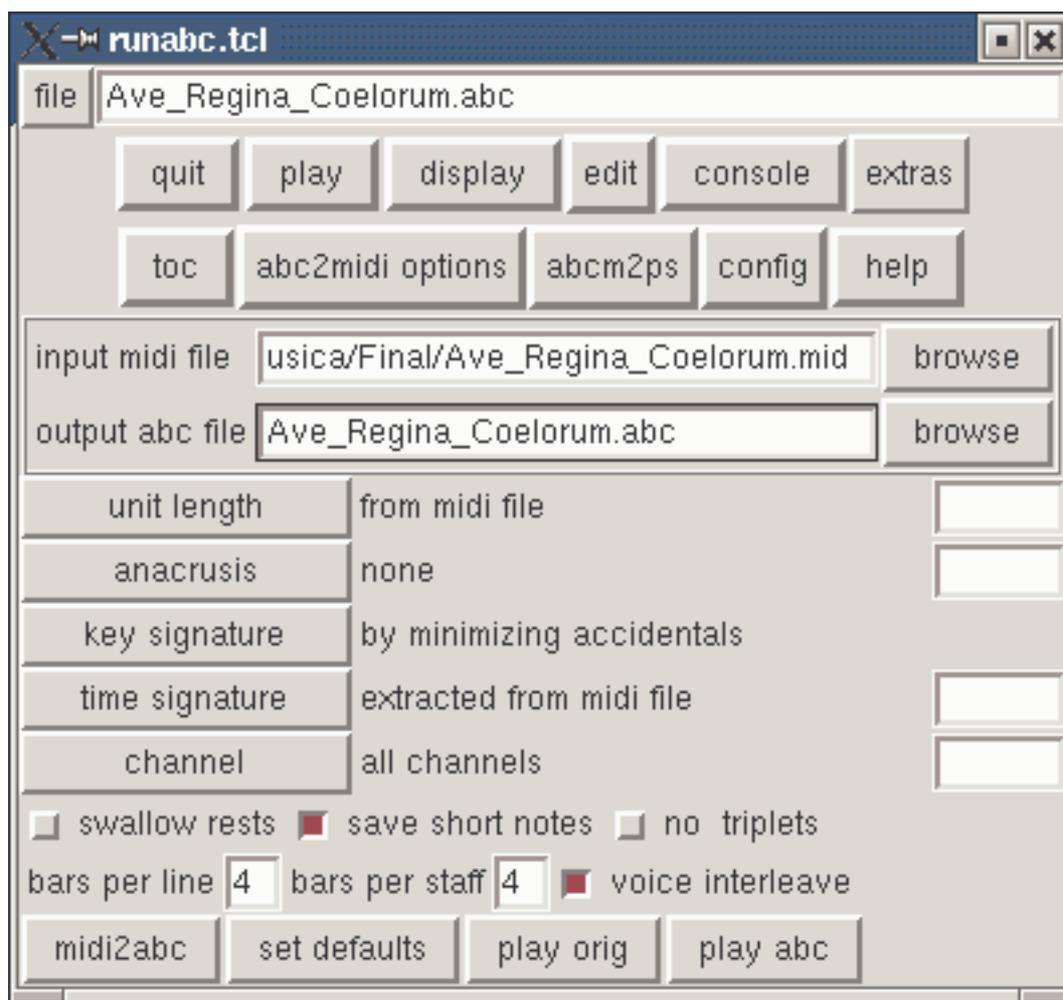
Die Befehlszeile ist wie folgt:

```
midi2abc file.mid -o file.abc
```

*midi2abc* besitzt viele Kommandozeilenparameter, auf die hier jedoch nicht eingegangen werden soll. Standardmäßig hat die erzeugte ABC PLUS nur einen Takt pro Zeile. Der beste Weg, um von einer MIDI-Datei ein ABC PLUS Quelltext zu erzeugen, ist *runabc.tcl* (siehe Abbildung unten). Es ist der Button Extras zu betätigen und dann ist *midi2abc* auszuwählen. Bei Auswahl der Option "Voice interleave" werden die Stimmen ineinander verschachtelt anstatt nacheinander ausgegeben. Über den Button „*midi2abc*“ wird die Konvertierung angestoßen.

Es dürfen keine perfekten Ergebnisse erwartet werden. Die Konvertierung von ABC PLUS Quelltexten nach MIDI funktioniert meist recht gut, die Gegenrichtung gelingt nicht immer. Triller werden z.B. als Folgen kurzer Noten dargestellt, Wiederholungen gibt es nicht im generierten ABC PLUS Quelltext, Notenlängen sind oft nicht korrekt im Sinne eines lesbaren Notenblattes...

Viele dieser Beschränkungen können auch in Zukunft nicht aufgehoben werden, aber es ist geplant in weiteren Versionen von *abc2midi* die Qualität der generierten ABC PLUS Quelltexte zu verbessern.



## 6.4 yaps

Dieses Programm ist ein weiterer Konverter, der ABC PLUS Quelltexte in Postscript konvertieren kann. Die Ausgabequalität ist jedoch nicht vergleichbar mit abcm2ps, so dass es hier nicht weiter betrachtet wird.

## 6.5 Unterschiede und Inkompatibilitäten

Leider sind abcm2ps und abc2midi nicht 100 prozentig kompatibel, weil abcm2ps eine gegenüber abc2midi erweiterte Syntax kennt. Viele spezielle Funktionen machen jedoch auch nur Sinn in gedruckten Notenblättern.

Wenn Musik mit ABC PLUS geschrieben wird sollte bzgl. der Konvertierbarkeit nach MIDI beachtet werden:

- abc2midi akzeptiert nur Q:-Felder wie z.B. Q:1/4=60 oder Q:200.
- Texte in doppelten Anführungszeichen sollten nur gültige Begleitakkordbezeichnungen enthalten.
- Wenn ein Wechsel des Notensystems in der Mitte eines Stückes vorgenommen wird, kommt abc2midi aus dem Tritt und generiert eine fehlerhafte MIDI Datei.
- Stimmteilung mittels & funktioniert nicht;
- weitere Inkompatibilitäten existieren sicherlich ....

Wegen dieser kleinen Inkompatibilitäten ist es schwierig, ein Stück so zu schreiben, dass es sowohl von abcm2ps als auch von abc2midi erfolgreich konvertiert werden kann. In der Theorie müssten eigentlich zwei Quelltexte geschrieben werden, eine Datei, die für die Notendarstellung optimiert ist und eine weitere, die mit abc2midi zum besten Ergebnis führt. Dies ist natürlich nicht akzeptabel. Hier kommt nun der abcpp Präprozessor ins Spiel, auf den im nächsten Kapitel eingegangen wird.

## 7 Teil VI Automatische Quelltext Umwandlung

### 7.1 *Der abcpp Präprozessor*

Ein Präprozessor ist ein Programm, welches eine Text Datei entsprechend den in der Datei enthaltenen Befehle. Abcpp ist ein Präprozessor welcher ausschließlich für ABC PLUS Dateien entworfen wurde. Er erlaubt

- Teile eines Stückes entsprechend von angegebenen Bedingungen ein- bzw. Auszuschließen
- Makros zu definieren, d.h. Symbole und Folgen nutzerdefinierter Befehle
- Befehle, Symbole und Noten umzubenennen
- Teile anderer Dateien einzubeziehen

Das abcpp ein Kommandozeilen-Programm ist, braucht nicht extra erwähnt zu werden. Es wird mit den Namen der Ausgangs- und Zieldatei(en) aufgerufen und ggf. zusätzlich definierten Symbolen.

#### 7.1.1 Grundsätzliche Benutzung

Wir werden zunächst ein Beispiel betrachten, bei dem eine ABC PLUS Datei geschrieben wird, die sich sowohl mit abcm2ps als auch mit abc2midi korrekt übersetzen lässt. Dieser Quelltext wird unter test.abp gespeichert:

```
X: 1
T: Test with abcpp
#ifdef ABCMIDI
T: (version for abc2midi)
Q: 1/4 = 120
#else
T: (version for abcm2ps)
Q: "Allegro" 1/4 = 120
#endif
K: C
cdef gabc'|c'bag fedc|
```

Zu beachten sind die Zeilen, die mit '#' beginnen. Diese Zeilen sind die Direktiven für den Präprozessor.

Die erste Direktive bedeutet: "Wenn das Symbol ABCMIDI definiert ist, dann ..." Wenn die Bedingung wahr ist, werden im Beispiel die nächsten beiden Zeilen prozessiert, anderenfalls werden die Zeilen die der #else-Direktive folgen, prozessiert. Die #endif Direktive beendet die Bedingung.

Um die Datei in einen Quelltext zu verwandeln, der von abc2midi akzeptiert wird, muss abcpp mit folgenden Parametern aufgerufen werden:  
abcpp -ABCMIDI test.abp test-midi.abc

Auf diesem Wege wird das Symbol ABCMIDI definiert und eine neue ABC PLUS Datei erzeugt.

```
X: 1
T: Test with abcpp
T: (version for abc2midi)
Q: 1/4 = 120
K: C
cdef gabc'|c'bag fedc|
```

Wenn abcpp ausgeführt wird, ohne dass symbole definiert werden, wird ein Quelltext passend für abcm2ps erzeugt:

```
abcpp test.abp test-ps.abc
```

```
X: 1
T: Test with abcpp
T: (version for abcm2ps)
Q: "Allegro" 1/4 = 120
K: C
cdef gabc'|c'bag fedc|
```

Nun soll ein weiteres Beispiel betrachtet werden. Einige ABC-Anwendungen unterstützen die Verwendung von unsichtbaren Pausen nicht. Um diese Art von Pausen trotzdem verwenden zu können, müssen folgende Zeilen in den Quelltext eingefügt werden:

```
#ifdef OLD
#define !x! z
#else
#define !x! x
#endif
```

Dies bedeutet : “Wenn das Symbol OLD definiert ist, wird !x! durch z ersetzt, sonst wird !x! durch x ersetzt“. Beim Schreiben des Quelltextes wird nun für unsichtbare Pausen die Notation !x! verwendet und durch abcpp je nachdem, ob OLD definiert ist oder nicht, in x oder z umgewandelt.

## 7.1.2 Weitergehende Möglichkeiten

In Kap. 4.4 wurde ein Beispiel einer Notation mit Systemwechsel im Stück dargestellt. Leider ist abc2midi nicht in der Lage mit variabler Anzahl von Stimmen umzugehen.

Unter Benutzung von abcpp kann jedoch ein Quelltext erstellt werden, mit dem beide Anwendungen korrekt arbeiten.

Die Idee ist, alle Stimmen auszuschreiben, auch die die nur Pausen enthalten. Zusätzlich werden Instruktionen für abcm2ps und abc2midi eingefügt. Das Ergebnis ist ein Quelltext, dessen Teile A und C in der Druckausgabe nur Stimme 3 enthalten während Teil B alle Stimmen enthält.

```
X: 1
T: Riu, riu, chiu, la guarda ribera!C: "Villancico" (Spain, XVIth
century)
M: C|
L: 1/2
Q: 1/2 = 240
#ifdef MIDI
P: ABCB
#endif
```

```

%%staves 3
V: 3 clef=treble-8 name="Tenor\nBass"
K: Am
% ONLY THE MEN
#ifdef MIDI
P: A
[V: 1] [M:none] z4 |z4z4 |z6 |
[V: 2] [M:none] z4 |z4z4 |z6 |
[V: 4] [K: Am octave=-1]\
[M:none] aaga |f2ed2efg|a2a2z2|
#endif
[V: 3] [M:none] AAGA |F2ED2EFG|A2A2z2|
w: Ri-u, ri-u, chi-u, la guar-da ri-be-ra!
%
#ifdef MIDI
[V: 1] z4 |z4z4 |z6 |
[V: 2] z4 |z4z4 |z6 |
[V: 4] aaga |f2eg2gef|d2d2z2|
#endif
[V: 3] AAGA |F2EG2GEF|D2D2z2|
w: Di\'os guar-d\'o el lo-bo de nue-stra cor-de-ra,
%
#ifdef MIDI
[V: 1] z4 |z4z4 |z4 |
[V: 2] z4 |z4z4 |z4 |
[V: 4] aaga |f2eg2gef|d2d2|
#endif
[V: 3] AAGA |F2EG2GEF|D2D2|
w: Di\'os guar-d\'o el lo-bo de nue-stra cor-de-ra.
% WOMEN AND MEN
#endif MIDI
%%staves [1 2 3 4]
V: 1 clef=treble name="S" sname="S"
V: 2 clef=treble name="A" sname="A"
V: 3 clef=treble-8 name="T" sname="T"
V: 4 clef=bass name="B" sname="B"
#else
P: B
#endif
[V: 1]AAGA|F2ED2EFG |A2A2z2|
w: Ri-u, ri-u, chi-u, la guar-da ri-be-ra!
[V: 2]FFEC|D2EF2EDD |C2C2z2|
[V: 3]cccG|A2AA2ADD |E2E2z2|
w: Ri-u, ri-u, chi-u, la guar-da ri-be-ra!
[V: 4]ffcf|d2Ad2c_BB|A2A2z2|
%
[V: 1] z4 |AAGA|F2EF2FEE|D2D2z2|
w: Di\'os guar-d\'o el lo-bo de nue-stra cor-de-ra,
[V: 2] z2EE |DCEC|D2CD2DCC|D2D2z2|
w: Di\'os guar-d\'o el lob', el lo-bo de nue-stra cor-de-ra,
[V: 3] ccBc |A2BA|A2AA2AAA|A2A2z2|
w: Di\'os guar-d\'o el lo-bo, el lo-bo de nue-stra cor-de-ra,
[V: 4] aaga |f2ef|d2Ad2dAA|d2d2z2|
%
[V: 1] z4 |AAGA|F2ED2DCC |D2D2z2 |
w: Di\'os guar-d\'o el lo-bo de nue-stra cor-de-ra.
[V: 2] z2EE|DCEC|D2CA,2A,A,A,|A,2A,2z2|

```

```

w: Di\'os guar-d\'o el lob', el lo-bo de nue-stra cor-de-ra.
[V: 3] ccBc|A2BA|A2AF2FEE |D2D2z2 |
w: Di\'os guar-d\'o el lo-bo, el lo-bo de nue-stra cor-de-ra.
[V: 4] aaga|f2ef|d2Ad2dAA |d2d2z2 |
% ONLY THE MEN
#ifdef MIDI
P: C
[V: 1] z4 |z8 |z4|z4 |
[V: 2] z4 |z8 |z4|z4 |
[V: 4] aaga|f2eg2gef|d4|aaga|
#else
%%staves 3
#endif
[V: 3] AAGA|F2EG2GEF|D4|AAGA|
w: El lo-bo ra-bio-so la qui-so mor-der, mas Di\'os po-de-%
#ifdef MIDI
[V: 1] z8 |z4|z4 |z8 |
[V: 2] z8 |z4|z4 |z8 |
[V: 4] f2feggef|d4|aaga|f2fedefg|
#endif
[V: 3] F2FEGGEF|D4|AAGA|F2FEDEFG|
w: ro-so la su-po de-fen-der; qui so-le ha-ce que no pu-die-sce pe-%
#ifdef MIDI
[V: 1] z4|z4 |z8 |z4 |
[V: 2] z4|z4 |z8 |z4 |
[V: 4] a4|aaga|f2feggef|d2d2|
#endif
[V: 3] A4|AAGA|F2FEGGEF|D2D2|
w: car: ni~aun o-ri-gi-nal e-sta Vir-gen no tu-vie-ra.

```

## 7.2 abc2abc

Dieses Programm ist Teil des abcMIDI Paketes. Es ist ein Kommandozeilen-Programm welches benutzt werden kann, um ABC PLUS Quelltexte auf verschiedene Weise zu modifizieren. abc2abc wird als Parameter der Name des zu modifizierenden Quelltextes und danach die Optionen angegeben:

- n <x> formatiert den Quelltext mit <x> Takten pro Zeile neu.
- t <n> transponiert die Musik um <n> Halntöne.
- d verdoppelt die Notenlängen
- v halbiert die Notenlängen.
- V <x> gibt nur Stimme <x> eines polyphonen Stückes aus
- X <n> In einer Datei mit mehreren Stücken werden die X:-Felder der Stücke beginnend mit <n> neu nummeriert

Wie üblich soll auch hier wieder ein Beispiel gegeben werden. Es soll eine Tonleiter modifiziert werden.

```

X: 1
L: 1/4
K: C
CDEF|GABc|cdef|gabc'|c'cCz|

```

Der folgende Aufruf von abc2abc transponiert die Tonleiter um 3 Halbtöne und gibt 2 Takte pro Zeile aus:

```
abc2abc cde.abc -n 2 -t 3
```

```
X:1
L:1/4
K:Eb
%
EFGA|Bcde|
efga|bc'd'e'|
e'eEz|
```

## 8 Teil VII Weitere Möglichkeiten

### 8.1 Nutzung von Notenblättern in anderen Programmen

Notenblätter im Postscript-Format können einfach in andere Formate konvertiert werden, die geeignet für Textverarbeitungsprogramme, Webseiten etc. sind. In der Praxis werden es vor allem zwei Formate sein: JPG and PNG, wobei letzteres zu bevorzugen ist.

Um Postscript zu PNG zu konvertieren, können Windows Benutzer das Programm GhostView benutzen. Unter dem Menüpunkt File/Convert ist als Device png16 zu wählen und die zu druckenden Seiten zu selektieren. Als Auflösung sei 300 dpi empfohlen.

Linux Benutzer können den GhostScript Interpreter direkt nutzen. Das folgende Script konvertiert eine Postscriptdatei zu einer PNG-Datei:

```
#!/bin/sh
FILE=$(basename $1 .ps)
gs -dNOPAUSE -q -dBATCHE -sPAPERSIZE=a4 \
-sDEVICE=pnggray \
-dTextAlphaBits=4 -dGraphicsAlphaBits=4 \
-r300x300 \
-sOutputFile=$FILE-%003d.png \
$1
```

Eine weitere Möglichkeit zur Konvertierung besteht in der Nutzung des Programmes convert aus dem ImageMagick Paket (<http://www.imagemagick.org/>). Es kann wie folgt verwendet werden:

```
convert -density 300x300 file.ps file.png
```

Der -density Parameter gibt die Auflösung an.

#### 8.1.1 Einfügen von Musik in L<sup>A</sup>T<sub>E</sub>X

Die englische Version dieses Tutorials wurde mit L<sup>A</sup>T<sub>E</sub>X geschrieben, welches als Textsatzsystem anstelle eines Textverarbeitungssystems verwendet werden kann.

Um mit ABC PLUS gesetzte Musik in L<sup>A</sup>T<sub>E</sub>X Dokumente einzufügen, muss zunächst festgelegt werden, ob das endgültige Format PDF oder Postscript sein soll.

In beiden Fällen müssen die Noten ins EPS (encapsulated Postscript) – Format umgewandelt werden. Dies wird entweder mit der Kommandozeilenoption -E von abcm2ps oder mit dem Script ps2epsi aus dem Ghostscript-Paket erreicht.

In das L<sup>A</sup>T<sub>E</sub>X-Dokument muss das graphicx-Paket und die EPS-Grafik eingebunden werden:

```
\documentclass[a4paper,12pt]{article}
\usepackage{graphicx}
\begin{document}
This is some ABC music:
\medskip
\includegraphics[width=\linewidth]{music.eps}
\end{document}
```

Wenn pdflatex benutzt werden soll, so müssen die Noten mit dem Programm epstopdf von EPS in das PDF-Format konvertiert werden, bevor sie im LATEX-Dokument eingebunden werden kann.

## **8.2 Umwandlung von Grafiken ins EPS-Format**

Oft liegen Grafiken als JPG, GIF oder PNG-Dateien vor. Um solche Grafiken mit dem %%EPS-Befehl in ABC PLUS Quelltexte einbinden zu können, müssen diese Dateien zunächst ins EPS-Format umgewandelt werden.

Dies kann mit dem Kommandozeilenprogramm bmeps getan werden, welches von der WEB-Seite <http://www.ctan.org/tex-archive/support/bmeps> heruntergeladen werden kann. Für Windows Nutzer steht ein statisch gelinktes EXE zur Verfügung.

bmeps wie in folgendem Beispiel benutzt:

```
bmeps -c myfile.png myfile.eps
```

Wenn die -c Option weggelassen wird, dann ist die resultierende EPS-Datei in Schwarz-Weiß.

## **8.3 Stimmen extrahieren**

Ein weiteres nützliches Werkzeug ist abc2prt, das einzelne Stimmen aus mehrstimmigen Stücken extrahieren kann. Durch das Programm wird ein neuer ABC PLUS Quelltext erzeugt, der die einzelne Stimme enthält.

Als Beispiel soll hier die Tenorstimme (Stimme 3) aus Ave Verum aus Kap. 5 extrahiert werden:

```
abc2prt -3 aveverum.abc aveverum-3.abc
```

Eine neue ABC PLUS Datei aveverum-3.abc wird erzeugt, welche nur die Stimme 3 enthält. Ac2prt ist ebenfalls in JEDABC integriert.

## **8.4 Grenzen von abcm2ps**

Obwohl abcm2ps ein sehr leistungsfähiges Programm ist, hat es derzeit noch einige Einschränkungen, die zu beachten sind:

- direkte Steuerung der Symbolpositionierung nicht möglich
- einige Formatierungsparameter können innerhalb eines Stückes nicht geändert werden.
- Der Nutzer kann nicht bestimmen, ob Balken über oder unter den Noten gezeichnet werden.
- Spezielle Notationen wie Perkussionsinstrumente oder gregorianische Chöre werden nicht unterstützt
- . . .

Die Liste war für die vorhergehenden Versionen um einiges länger... Die fehlenden Punkte werden in zukünftigen Versionen sicherlich implementiert.

## **8.5 Abschließende Bemerkungen**

Diese Anleitung wurde von Guido Gonzato, [ggonza@tin.it](mailto:ggonza@tin.it) geschrieben, von Frank-Michael Krause ([mich.krause@web.de](mailto:mich.krause@web.de)) ins Deutsche übersetzt. Sie wird unter der GNU GPL Lizenz veröffentlicht. Dies bedeutet, dass diese Anleitung kostenlos benutzt und frei verteilt werden kann. Wenn der Text modifiziert wird, muss er öffentlich verfügbar gemacht werden. Für Hinweise, Fehler-Reports, Vorschläge und Kommentare sind Autor und Übersetzer offen und dankbar.

Ein großes „Danke Schön“ an den Autor von abcm2ps, Jean-Francois Moine, der es geschafft hat, ein sehr nützliches Programm zu erstellen. Danke auch an Michael Methfessel, der die ursprüngliche Version von abc2ps geschrieben hat und an James Allwright, dem Autor von abcMIDI. Ebenfalls Dank an Seymour Schlien für die Pflege des Programmes abcMIDI und an Norman Schmidt, der geholfen hat, diese Anleitung ins Englische zu übersetzen. Vielen Dank auch an meinen Freund Maestro Sandro Pasqualetto und an Gianni Cunich für ihre Vorschläge zur Verbesserung dieser Anleitung und nicht zuletzt an all die Menschen welche zu ABC PLUS beigetragen haben!

### **8.5.1 Bitte eine kleine Spende**

Diese Anleitung ist kostenlos. Es ist jedoch viel Zeit in das Schreiben, entfehlen und verbessern der Anleitung geflossen. Für eine kleine Spende wäre der Autor dankbar, es wäre auch ein Anreiz für weitere Verbesserungen an dieser Anleitung.

Wenn sie einige Euro oder US Dollars erübrigen können und diese Anleitung nützlich finden, senden Sie bitte Ihre Spende an meine Adresse:

Guido Gonzato, Via Monte Ortigara 19/a, 37127 Verona (Italy).

Oder überweisen sie auf mein Konto:

Unicredito Italiano, c.c. 8597566, ABI 2008, CAB 11770.

Falls Sie kein Geld erübrigen können, würde ich mich über eine Postkarte freuen. Ich freue mich besonders über Landschaftsmotive für meine Bürowände. Vielen Dank!

### **8.5.2 Im Gedenken an Annarosa Del Piero, 1930–2000**

Ich hatte das Privileg, ein Freund von Annarosa zu sein. Ohne diese Freundschaft wäre ich ein anderer Mensch.

Annarosa war einmalig. Sie verband eine tiefe Liebe mit Kunst und Musik. Sie teilte ihre Liebe mit mir, als ich noch ein Kind war, indem sie mir Operaufnahmen schenkte. Sie fuhr mit mir mit dem Zug das erste Mal in meinem Leben nach Venedig und sie zeigte mir die Schönheit der Berge.

Sie bekämpfte ihre Krankheit mit Courage und Würde. Bis zum Ende hörte sie ihre Lieblingsmusik und gab mir wundervolle Aufnahmen von Opern als Geschenk. Diese Anleitung ist dem Gedenken an sie gewidmet.

Ciao, Annarosa.

## Teil VIII Anhang

### Anhang A **ABC PLUS Felder**

Feld	Position	Bedeutung und Beispiel
A:	Kopf	Gebiet (Area). A:Liverpool
B:	Kopf	Buch. B:Groovy Songs
C:	Kopf	Komponist(Composer). C:The Beatles
D:	Kopf	Discographie. D:The Beatles Complete Collection
d:	Rumpf	Dekorationen. d:!pp! * * !mf! * !ff!
F:	Kopf	Datei(File). F:http://www.beatles.org/help.abc
G:	Kopf	Gruppe. G:guitar
H:	Kopf	History. H:This son was written...
I:	Kopf	Information. I:lowered by a semitone
K:	zuletzt im Kopf	Tonart (Key). K:C
L:	Kopf, Rumpf	Standard Noten Länge. L:1/4
M:	Kopf, Rumpf	Taktart (Metre). M:3/4
N:	Kopf	Notizen :See also...
O:	Kopf	Original. O:English
P:	Kopf, Rumpf	Teil (Part). P:Start
Q:	Kopf, Rumpf	Tempo. Q:1/2=120
R:	Kopf	Rhythmus. R:Reel
S:	Kopf	Quelle (Source). S:Collected in Liverpool
T:	an 2. Stelle im Kopf	Titel. T:Help!
U:	Kopf	Nutzer definiert (User defined). U:T=!trill!
V:	Kopf, Rumpf	Stimme (Voice). V:1
W:	Rumpf	Liedtext am Ende (Lyrics). W:Help! I need...
w:	Rumpf	Liedtext unter jeder Zeile. w:Help! I need...
X:	Beginn des Kopfes	Index Nummer. X:1
Z:	Kopf	Transkriptions Notizen. Z:Transcribed by ear

### Anhang B **Glossar**

- editor:** ein Programm um 'ASCII text' zu schreiben, d.h. ohne spezielle Formatierung, z.B. Windows' Notepad (ugh), emacs oder vim.
- Schriftart (Font):** Erscheinungsbild der Buchstaben, z.B. Times oder Helvetica.
- GPL:** eine Software Lizenz, die für viele Programme verwendet wird, die frei über das Internet verfügbar ist. Kurz gesagt können unter GPL stehende Programme frei benutzt und modifiziert werden. Details sind unter <http://www.gnu.org/> zu finden.
- MIDI:** grob gesprochen, das hörbare Gegenstück zum Notenblatt. MIDI-Dateien können mit speziellen Programmen (MIDI-Player) wiedergegeben werden.

- PDF:** Ein Datei Format das von Adobe erfunden wurde und zur Geräte- und Systemunabhängigen Kodierung von Dokumenten verwendet wird. Es ist eine Weiterentwicklung von Postscript.
- POSTSCRIPT:** Dateiformat, das von Adobe erfunden wurde; Vektorformat

## Anhang C Zeichensätze

Eins der schönsten Dinge auf der Welt ist die Vielfältigkeit. Verschiedene Sprachen verwenden verschiedene Buchstaben. Dies kann jedoch auch zu Kompatibilitätsproblemen führen, wenn z.B. ein italienischer Musiker seinem amerikanischen Kollegen oder Kollegin einen ABC PLUS Quelltext schickt. Glücklicher Weise gibt es einen Zeichensatz ISO 8859-1 (Latin1) welcher die Sonderzeichen vieler Sprachen enthält: die aller Zentraleuropäischen Länder und die der englischsprechenden Länder. Diese Zeichen haben ASCII-Codes zwischen 160 und 255 und können als Folgen `\xxx` gesetzt werden, wenn sie nicht auf der Tastatur verfügbar sind.

Die Latin1 Zeichen und die zugehörigen Oktal-Codes in der nachfolgenden Tabelle dargestellt. Es sei empfohlen, dass die `\xxx` Codierungen auch verwendet werden wenn die Zeichen auf dem Keyboard verfügbar sind, weil dies den Quelltext portabel macht.

### ISO 8859-1 (Latin1)

\240	ı \241	ç \242	£ \243	¤ \244	¥ \245	\246	§ \247
¨ \250	© \251	ª \252	« \253	¬ \254	- \255	® \256	ˆ \257
° \260	± \261	² \262	³ \263	´ \264	µ \265	¶ \266	· \267
¸ \270	¹ \271	º \272	» \273	¼ \274	½ \275	¾ \276	¿ \277
À \300	Á \301	Â \302	Ã \303	Ä \304	Å \305	Æ \306	Ç \307
È \310	É \311	Ê \312	Ë \313	Ì \314	Í \315	Î \316	Ï \317
Ð \320	Ñ \321	Ò \322	Ó \323	Ô \324	Õ \325	Ö \326	× \327
Ø \330	Ù \331	Ú \332	Û \333	Ü \334	Ý \335	Þ \336	ß \337
à \340	á \341	â \342	ã \343	ä \344	å \345	æ \346	ç \347
è \350	é \351	ê \352	ë \353	ì \354	í \355	î \356	ï \357
ð \360	ñ \361	ò \362	ó \363	ô \364	õ \365	ö \366	÷ \367
ø \370	ù \371	ú \372	û \373	ü \374	ý \375	þ \376	ÿ \377

## Anhang D Formatierungsbefehle

### D1 Seitenformat

Mit den folgenden Befehlen kann das Seitenlayout geändert werden:

**%%pageheight <length>** : Setzt die Seitenhöhe auf <length> . Für europäisches A4 Papier ist der korrekte Wert 29,7cm; für US Letter 11 in (in=Zoll).

**%%pagewidth <length>** : Setzt die Seitenbreite auf <length> . Für europäisches A4 Papier ist der korrekte Wert 21 cm, für US Letter 8.5in.

**%%topmargin <length>** : Setzt den oberen SeitenRand auf <length> .

**%%botmargin <length>** : Setzt den unteren Seitenrand auf <length> .  
**%%leftmargin <length>** : Setzt den linken Seitenrand auf <length> .  
**%%rightmargin <length>** : Setzt den rechten Seitenrand auf <length> .  
**%%indent <length>** : Set die Einrückung der ersten Zeile auf <length> .  
**%%staffwidth <length>** : kann alternativ zu %%pageheight und %%pagewidth.  
**%%footer <text>** : Setzt den angegebenen Text als Fußzeile auf jede Seite.  
**%%header <text>** : Setzt den angegebenen Text als Kopfzeile auf jede Seite.  
**%%landscape <boolean>** : wenn boolean=1, wird Querformat verwendet.

## D2 Text

Diese Befehle können benutzt werden, um Textzeilen im Notenblatt einzufügen. Die Befehle zum Setzen der Schriftart und der Abstände werden später aufgeführt.

**%%text <text>** : gibt <text> aus, z.B.:  
 %%text Spanish folk song

**%%center <text>** : wie oben, jedoch zentriert.

**%%begintext...%%endtext** : Dieses Befehlspaar schließt einen Textblock ein, wenn dem %% kein Text folgt, wird ein Absatz beendet.

Beispiel:

```
%%begintext
%% Spanish folk song, usually
%% accompanied by guitar and cimbals.
%%endtext
```

Dem Befehl %%begintext kann ein Parameter zur Festlegung der Formatierung angefügt werden:

%%begintext obeylines	- gibt den Text ohne spezielle Formatierung aus.
%%begintext fill	- formatiert and justiert den Text
%%begintext justify	- richtet den rechten Rans aus
%%begintext skip	- die folgenden Zeilen werden ignoriert

**%%sep**: Erzeugt einen zentrierten Separator (eine kurze Linie)

**%%sep <h1> <h2> <l>** : Erzeugt einen Separator der Länge <l> mit einem Abstand von <h1> darüber und <h2> darüber.

**%%vskip <h>** : Fügt einen vertikalen Abstand ein

**%%newpage**: Erzeugt einen Seitenumbruch

## D3 Schriftarten

Diese Befehle werden benutzt um die Schriftarten festzulegen, die in den verschiedenen Teilen des Notenblattes verwendet werden.

ZU beachten ist, dass die üblichen Windows True Type Fonts sind nicht die Schriftarten, die von abcm2ps benutzt werden. Abcm2ps benutzt Postscript-Schriftarten, die von GhostScript bereitgestellt werden. Die Standard Ghostscript Schriftarten sind in Anhang F dargestellt.

Wie neue Schriftarten hinzugefügt werden können wurde in Kap. 5.11.4 ausführlich erläutert.

**%%font <string>** : Deklaration einer Schriftart für die spätere Benutzung  
**%%titlefont <string>** : Schriftart für die erste Titelzeile (erstes T:-Feld)  
**%%subtitlefont <string>** : Schriftart des Untertitels (zweites T:-Feld)  
**%%composerfont <string>** : Schriftart für Komponist (C:-Feld)  
**%%partsfont <string>** : P:-Feld Schriftart  
**%%tempofont <string>** : tempo Schriftart.  
**%%gchordfont <string>** : Schriftart für Gitarren Akkorde.  
**%%infofont <string>** : text Schriftart in I:-Feldern.  
**%%textfont <string>** : text Schriftart in %%text - Zeilen.  
**%%vocalfont <string>** : Schriftart der Texte in w:-Zeilen  
**%%footerfont <string>** : Fußzeilen-Schriftart  
**%%headerfont <string>** : Kopfzeilen-Schriftart  
**%%wordsfont <string>** : Schriftart der Texte in W:-Zeilen  
**%%titlecaps <Boolean>** : wenn Boolean = 1, wird er Titel in Großbuchstaben geschrieben  
**%%titleleft <Boolean>** : wenn Boolean = 1 wird der Titel linksbündig statt zentriert ausgegeben

## D4 Abstände

Die folgenden Befehle bestimmen die Abstände zwischen den Elementen des Notenblattes:

**%%topspace <length>** : Setzt den vertikalen Abstand des Stücken vom Seitenbeginn.  
Achtung, ein Stück kann mit %%text Befehlen vor dem Titel beginnen  
**%%titlespace <length>** : Setzt den Vertikalen Abstand über dem Titel.  
**%%subtitlespace <length>** : Setzt den vertikalen Abstand über dem Untertitel  
**%%composerspace <length>** : Setzt den vertikalen Abstand über der Komponistenangabe  
**%%musicospace <length>** : Setzt den vertikalen Abstand über der ersten Notenzeile  
**%%partsspace <length>** : Setzt den vertikalen Abstand über einem neuen Teil  
**%%vocalspace <length>** : Setzt den vertikalen Abstand zwischen Notenzeile und Liedtext  
**%%wordsspace <length>** : Setzt den vertikalen Abstand über dem Liedtext am Ende des Stückes  
**%%textspace <length>** : Setzt den Abstand über Texten  
**%%infospace <length>** : Setzt den Abstand über Infozeilen  
**%%staffsep <length>** : Setzt den Abstand zwischen verschiedenen Notensystemen  
**%%sysstaffsep <length>** : Setzt den Abstand zwischen den Notenzeilen eines Systems  
**%%barsperstaff <Anzahl>** : Versuche pro Zeile <Anzahl> Takte auszugeben  
**%%parskipfac <Faktor>** : Setzt den Faktor für den Abstand zwischen Teilen  
**%%lineskipfac <Faktor>** : Setzt den Faktor für den Abstand zwischen den Textzeilen

**%%stretchstaff <Boolean>** : Nicht gefüllte Zeilen werden über die Seitenbreite gestreckt wenn Boolean=1

**%%stretchlast <Boolean>** : Nur das letzte System wird gestreckt

**%%maxshrink <Faktor>** : Setzt den Faktor (zwischen 0.0 und 1.0), wie stark die Noten horizontal komprimiert werden können, wenn die Zeilenumbrüche automatisch bestimmt werden.

**%%scale <float>** : Setzt den Skalierungsfaktor

**%%slurheight <float>** : setzt den Faktor für die Höhe der Bindebögen, Faktoren kleiner 1 erzeugen flachere Bögen, Werte größer 1 erhöhen die Bögen

**%%notespacingfactor <float>** : Setzt den Proportionalabstand der Noten, Standard ist 1.414 (Wurzel aus 2) . Der Wert 1 macht alle Abstände gleich.

## D5 Andere Befehle

**%%autoclef <bool>** : wenn 0, wird der automatische Wechsel des Schlüssels verhindert, wenn die Töne zu hoch oder zu tief werden.

**%%deco <str> <int> <str> <int> <int> <int> <str>** : definiert eine neue Dekoration, Siehe Kapitel 5.11.3

**%%flatbeams <bool>** : wenn 1, erzwingt flache Balken für Dudelsack-Noten (K:HP).

**%%freegchord <bool>** : wenn 1, verhindert, dass die Zeichen # , b und = als Kreuz, b und Auflösungszeichen in Gitarren Akkorden ausgegeben werden. Wenn gesetzt, müssen Vorzeichen mit Escape-Zeichen erzeugt werden (z.B. \#)

**%%postscript <str>** : Eine Folge dieses Befehls kann genutzt werden, um neue Postscript Routinen zu definieren oder bestehende zu ändern.

**%%measurefirst <int>** : Die Numerierung der Takte beginnt mit <int>

**%%measurenb <int>** : Alle <int> Takte wird eine Taktnumerierung ausgegeben

**%%EPS <str>** : Eine externe EPS Datei wird eingefügt

**%%measurebox <bool>** : Zeichnet eine Box um die Taktnummern

**%%partsbox <bool>** : Zeichnet eine Box um Name der Teile

**%%encoding <int>** : (für Experten) Setzt die Zeichensatzkodierung auf SO-Latin<int>, wobei <int> zwischen 0 und 6 liegen kann.

**%%continueall <bool>** : wenn gesetzt werden Zeilenumbrüche ignoriert

**%%exprabove <bool>** : Ausdruckssymbole werden über dem Notensystem ausgegeben. Wenn weder %%exprabove noch %%exprbelow gesetzt wurde, werden die Symbole über dem System ausgegeben, wenn Liedtext vorhanden ist, andernfalls darunter. %%exprabove hat Vorrang vor %%exprbelow.

**%%exprbelow <bool>** : Ausdruckssymbole werden unter dem Notensystem ausgegeben

**%%graceslurs <bool>** : An Vorschlägen werden Bindebögen ausgegeben

**%%infoline <bool>** : Rhythmus und Original werden in der selben Zeile ausgegeben

**%%musiconly <bool>** : wenn gesetzt, wird kein Liedtext ausgegeben

**%%printtempo <bool>** : wenn =0 wird keine Tempoangabe ausgegeben

**%%straightflags <bool>** : Gerade Fähnchen an den Notenhälsen bei Dudelsackmusik

**%%oneperpage <bool>** : Ein Lied pro Seite

**%%squarebreve <bool>** : ‘brevis’ Noten werden im rechteckigen Format ausgegeben  
**%%vocalabove <bool>** : Liedtexte werden über der Notenzeile ausgegeben  
**%%withxrefs <bool>** : Gibt die X:-Feld Nummer im Titel aus  
**%%writehistory <bool>** : Gibt Noten, History usw. Aus

## D6 abcMIDI Befehle

Einige der folgenden Befehle können nur von Nutzern verstanden werden, die bereits Erfahrung mit dem Aufbau von MIDI Dateien haben.

**%%MIDI channel <n>** : Wählt den MIDI Kanal aus  
**%%MIDI drumon** : Schaltet die Schlagzeugbegleitung ein  
**%%MIDI drumoff** : Schaltet die Schlagzeugbegleitung aus  
**%%MIDI program [c] <n>** : setzt das Instrument <n> für Kanal [c]  
**%%MIDI beat <a> <b> <c> <d>** : steuert die Lautstärke der Noten innerhalb eines Taktes. Dieser Befehl kann als alternativer Weg benutzt werden, um die Dynamik des Stückes zu steuern (setzen von fff, pp, ...)  
**%%MIDI beatstring <string>** : ähnlich %%MIDI beat, verwendet jedoch jedoch eine fmp-Folge  
**%%MIDI transpose <n>** : Transponiert die Ausgabe um <n> Halbtöne.  
**%%MIDI rtranspose <m>** : Transponiert relativ zu einem vorher ausgeführten %%transpose-Befehl  
**%%MIDI c <n>** : Definiert die MIDI-Note, die dem c entspricht. Standard ist 60.  
**%%MIDI grace <float>** : Bestimmt den Bruchteil der folgenden Note, der bei Vorschlägen benutzt wird, z.B. 1/6  
**%%MIDI chordname <string n1 n2 n3 n4 n5 n6>** : Definition von neuen Akkorden bzw. Änderung von bereits definierten Akkorden. Siehe Kapitel 6.1.4.  
**%%MIDI chordprog <n>** : Setzt das MIDI Instrument für Begleitakkorde (0...127)  
**%%MIDI bassprog <n>** : Setzt das MIDI Instrument für die Bass Noten (0...127)  
**%%MIDI chordvol <n>** : Setzt die Lautstärke der Akkordnoten (0...127)  
**%%MIDI bassvol <n>** : Setzt die Lautstärke der Bassnoten (0–127).  
**%%MIDI gchordon** : Schaltet die Akkordbegleitung an  
**%%MIDI gchordoff** : Schaltet die Akkordbegleitung aus  
**%%MIDI drum <string> <programs> <velocities>** : Definition des Schlagzeug-Rhythmus, siehe Kapitel 6.1.5  
**%%MIDI control <bass/chord> <n1 n2>** : Erzeugt ein MIDI control Ereignis. Wenn “bass” oder “chord” angegeben ist, so wird das Ereignis für den Bass- oder Akkord-MIDI-Kanal generiert, ansonsten für den Melodie-MIDI-Kanal. <n1> ist die MIDI Control Nummer (0...127) , <n2> ist der Wert.  
**%%MIDI pitchbend <bass/chord> <n1 n2>** : Erzeugt ein Pitchbend-ereignis für den aktuellen MIDI-Kanal, oder für den Bass- oder Akkord-MIDI-Kanal. Die beiden Werte geben die Tonhöhenänderung an.

**%%MIDI nobarlines:** bewirkt, dass ein Vorzeichen nur für die folgende Not gilt und nicht bis zum Taktende

**%%MIDI barlines:** Deaktiviert %%nobarlines.

**%%MIDI ratio < n1 n2 > :** Setzt das Verhältnis der Notenlängen bei gebrochenen Rhythmen. Normalerweise macht c>c die erste Note dreimal so lange wie die zweite. Dieses Verhältnis kann geändert werden, z.B. %%ratio 2 1

## **Anhang E Postscript Schriftarten**

Es gibt 35 standard POSTSCRIPT Schriftarten. Bis auf ZapfDingbats (welche nicht von abcm2ps unterstützt wird) sind sie im folgenden alle aufgelistet.

<b>Bookman-Demi</b>	<i>Helvetica-Narrow-Oblique</i>
<b>Bookman-DemiItalic</b>	<b>Helvetica-Narrow-Bold</b>
Bookman-Light	<b>Helvetica-Narrow-BoldOblique</b>
<i>Bookman-LightItalic</i>	Palatino-Roman
Courier	<i>Palatino-Italic</i>
<i>Courier-Oblique</i>	<b>Palatino-Bold</b>
<b>Courier-Bold</b>	<i>Palatino-BoldItalic</i>
<b>Courier-BoldOblique</b>	NewCenturySchlbk-Roman
AvantGarde-Book	<i>NewCenturySchlbk-Italic</i>
<i>AvantGarde-BookOblique</i>	<b>NewCenturySchlbk-Bold</b>
<b>AvantGarde-Demi</b>	<b><i>NewCenturySchlbk-BoldItalic</i></b>
<b>AvantGarde-DemiOblique</b>	Times-Roman
Helvetica	<i>Times-Italic</i>
<i>Helvetica-Oblique</i>	<b>Times-Bold</b>
<b>Helvetica-Bold</b>	<b><i>Times-BoldItalic</i></b>
<b>Helvetica-BoldOblique</b>	Σψμβολ
Helvetica-Narrow	<i>ZapfChancery-MediumItalic</i>

## **Anhang F MIDI Instrumente**

Die folgende Tabelle enthält eine komplette Liste aller General MIDI Instrumente unterteilt nach Instrumentenfamilien. Es sein hier nochmals daran erinnert, das für abc2MIDI die Nummer um 1 verringert werden muss.

**Piano**

1. Acoustic Grand
2. Bright Acoustic
3. Electric Grand
4. Honky-Tonk
5. Electric Piano 1
6. Electric Piano 2
7. Harpsichord
8. Clavinet

**Guitar**

25. Nylon String Guitar
26. Steel String Guitar
27. Electric Jazz Guitar
28. Electric Clean Guitar
29. Electric Muted Guitar
30. Overdriven Guitar
31. Distortion Guitar
32. Guitar Harmonics

**Ensemble**

49. String Ensemble 1
50. String Ensemble 2
51. SynthStrings 1
52. SynthStrings 2
53. Choir Aahs
54. Voice Oohs
55. Synth Voice
56. Orchestra Hit

**Pipe**

73. Piccolo
74. Flute
75. Recorder
76. Pan Flute
77. Blown Bottle
78. Skakuhachi
79. Whistle
80. Ocarina

**Synth Effects**

97. FX 1 (rain)
98. FX 2 (soundtrack)
99. FX 3 (crystal)
100. FX 4 (atmosphere)
101. FX 5 (brightness)
102. FX 6 (goblins)
103. FX 7 (echoes)
104. FX 8 (sci-fi)

**Sounds Effects**

121. Guitar Fret Noise
122. Breath Noise
123. Seashore
124. Bird Tweet
125. Telephone Ring
126. Helicopter
127. Applause
128. Gunshot

**Chromatic Percussion**

9. Celesta
10. Glockenspiel
11. Music Box
12. Vibraphone
13. Marimba
14. Xylophone
15. Tubular Bells
16. Dulcimer

**Bass**

33. Acoustic Bass
34. Electric Bass(finger)
35. Electric Bass(pick)
36. Fretless Bass
37. Slap Bass 1
38. Slap Bass 2
39. Synth Bass 1
40. Synth Bass 2

**Brass**

57. Trumpet
58. Trombone
59. Tuba
60. Muted Trumpet
61. French Horn
62. Brass Section
63. SynthBrass 1
64. SynthBrass 2

**Synth Lead**

81. Lead 1 (square)
82. Lead 2 (sawtooth)
83. Lead 3 (calliope)
84. Lead 4 (chiff)
85. Lead 5 (charang)
86. Lead 6 (voice)
87. Lead 7 (fifths)
88. Lead 8 (bass+lead)

**Ethnic**

105. Sitar
106. Banjo
107. Shamisen
108. Koto
109. Kalimba
110. Bagpipe
111. Fiddle
112. Shanai

**Organ**

17. Drawbar Organ
18. Percussive Organ
19. Rock Organ
20. Church Organ
21. Reed Organ
22. Accordion
23. Harmonica
24. Tango Accordion

**Solo Strings**

41. Violin
42. Viola
43. Cello
44. Contrabass
45. Tremolo Strings
46. Pizzicato Strings
47. Orchestral Strings
48. Timpani

**Reed**

65. Soprano Sax
66. Alto Sax
67. Tenor Sax
68. Baritone Sax
69. Oboe
70. English Horn
71. Bassoon
72. Clarinet

**Synth Pad**

89. Pad 1 (new age)
90. Pad 2 (warm)
91. Pad 3 (polysynth)
92. Pad 4 (choir)
93. Pad 5 (bowed)
94. Pad 6 (metallic)
95. Pad 7 (halo)
96. Pad 8 (sweep)

**Percussive**

113. Tinkle Bell
114. Agogo
115. Steel Drums
116. Woodblock
117. Taiko Drum
118. Melodic Tom
119. Synth Drum
120. Reverse Cymbal

## Anhang G Schlaginstrumente

Diese Instrumente können mit dem Befehl %%MIDI drum benutzt werden bzw. im MIDI Kanal 10. Die Angegebenen Nummern sind Notennummern.

35.	B,,,	Acoustic Bass Drum	36.	C,,	Bass Drum 1	37.	^C,,	Side Stick
38.	D,,	Acoustic Snare	39.	^D,,	Hand Clap	40.	E,,	Electric Snare
41.	F,,	Low Floor Tom	42.	^F,,	Closed Hi Hat	43.	G,,	High Floor Tom
44.	^G,,	Pedal Hi-Hat	45.	A,,	Low Tom	46.	^A,,	Open Hi-Hat
47.	B,,	Low-Mid Tom	48.	C,	Hi Mid Tom	49.	^C,	Crash Cymbal 1
50.	D,	High Tom	51.	^D,	Ride Cymbal 1	52.	E,	Chinese Cymbal
53.	F,	Ride Bell	54.	^F,	Tambourine	55.	G,	Splash Cymbal
56.	^G,	Cowbell	57.	A,	Crash Cymbal 2	58.	^A,	Vibraslap
59.	B,	Ride Cymbal 2	60.	C	Hi Bongo	61.	^C	Low Bongo
62.	D	Mute Hi Conga	63.	^D	Open Hi Conga	64.	E	Low Conga
65.	F	High Timbale	66.	^F	Low Timbale	67.	G	High Agogo
68.	^G	Low Agogo	69.	A	Cabasa	70.	^A	Maracas
71.	B	Short Whistle	72.	c	Long Whistle	73.	^c	Short Guiro
74.	d	Long Guiro	75.	^d	Claves	76.	e	Hi Wood Block
77.	f	Low Wood Block	78.	^f	Mute Cuica	79.	g	Open Cuica
80.	^g	Mute Triangle	81.	a	Open Triangle			