Typesetting music with PMX

by

Cornelius C. Noack

noack@itp.uni-bremen.de

— Version 2.61 / June 2011
(PMX features up to version 2.613 included)
Acknowledgement

This tutorial owes its very existence to the work by Luigi Cataldi, who a few years ago produced a wonderful manual for PMX in Italian. Luigi’s manual features many examples which help greatly in understanding some of the arguably arcane PMX notation.

Even though the Cataldi manual is, as Don Simons has aptly remarked, “written in the language of music”, it nevertheless seemed useful to have access to it for non-Italian speakers, so Don asked around for help on a ‘retranslation’.

In fact, that is what the present tutorial started out with: essentially a retranslation of the PMX part of Luigi’s manual back into English, using, where that seemed feasible, Don’s original PMX manual. I had been thinking for some time of producing some examples (and an index) for the updated (PMX 2.40) version of that manual, and now, taking Luigi’s Italian version as a basis, it seemed an easy thing to do.

Of course, as such projects go: soon after the first version had appeared in 2002, it tended to get out of hand — Don Simons actively produced one new beta version of PMX after the other, and I simply could not keep up with his pace.

So alas: 5 long years went by before the first update of the tutorial – reflecting all PMX changes from Version 2.40 to Version 2.514, in one giant step! – had become possible.

But, incessantly, as the development of modern TeX and LaTeX has moved on in an accelerating pace, and so have MusiXTeX and PMX. This tutorial was also bound to follow suite. So now the current status of PMX (2.613) and this tutorial are closely synchronized for the first time.

After thanking Luigi, in particular for his kind permission to use his version of the manual and its examples liberally, it behooves me to thank Don even more, not only for originating and continuously maintaining and improving PMX; his suggestions and continuous assistance in my work on this tutorial throughout the years were invaluable, as was his conscientious reading and improving, from the first draft to the present version. Many friends (too numerous to mention all of them) kept me busy trying to keep up with their productive suggestions and clarifications on the TeX-music users’ list during the past years.

In memoriam Daniel Taupin
Contents

A Introduction
   How to use this tutorial ............................................ 1
   A 1 \TeX, \textsc{Musix}\TeX, \textsc{PMX}, \textsc{M-Tx} .......................... 2
   A 2 Installation ..................................................... 4
   A 3 Authors of the Software .................................... 6
   A 4 The Werner Icking Music Archive .............................. 7

B A PMX Tutorial .................................................. 9
   Conventions for this tutorial .................................... 9
   B 1 Running \textsc{PMX} ............................................ 10
      B 1.1 Concatenating several files ............................. 13
   B 2 Preliminary Concepts ....................................... 15
   B 3 Preamble ...................................................... 15
      B 3.1 Numerical input .......................................... 16
   Body of the Input File ........................................... 22
   B 4 Commands for the Individual Staves ......................... 25
      B 4.1 Notes ...................................................... 25
      B 4.2 Dotted Notes .............................................. 26
      B 4.3 Stems ..................................................... 27
      B 4.4 Other Note Parameters ................................... 27
      B 4.5 Rests ...................................................... 29
      B 4.6 Xtuplets .................................................. 31
      B 4.7 Chords .................................................... 34
         B 4.7.1 Arpeggio .............................................. 35
      B 4.8 Grace notes ................................................ 36
      B 4.9 Ornaments ................................................ 38
      B 4.10 Beams .................................................... 40
      B 4.11 Slurs and Ties .......................................... 44
         Pick a package ................................................ 44
         B 4.11.1 General slur usage ................................ 46
         B 4.11.2 Invoking and using Type K slurs .................. 48
         B 4.11.3 Use of Type M Slurs and Ties ..................... 51
         B 4.11.4 Special considerations for font-based slurs .... 52
CONTENTS

D Limitations, error messages, and bugs 96
  D 1 Limitations ................................................. 96
  D 2 PMX’s error messages ..................................... 98
  D 3 Bugs ......................................................... 98
    D 3.1 A Benign Bug ............................................ 98

E Tricks of the Trade 99
  E 1 Simple tricks ............................................... 99
    E 1.1 Special coding in L’Incoronazione di Poppea .......... 99
    E 1.2 Text after final system .................................. 99
    E 1.3 Clef octaviation ......................................... 100
  E 2 More tricks .................................................. 101
    E 2.1 Changing vertical positioning of instrument name .. 101
    E 2.2 Xtuplets ending with a rest .............................. 102
    E 2.3 Shorthand notation for consecutive quavers .......... 104
    E 2.4 Varying the stave sizes ................................. 105
    E 2.5 Stuff in front of the clefs of the first system ..... 107

F An Extension of PMX: M-Tx 109

G Appendix: Examples 110
  G 1 Dons Example Files ......................................... 110
  G 2 Full-score examples ......................................... 110
    G 2.1 Dufay, Kyrie (PMX code) : .............................. 111
    G 2.2 Vivaldi, Mundi Rector (M-Tx code) : ................. 114
    G 2.3 Caccini, Amor l’ali m’impenna (M-Tx code) .......... 118

Index 124
List of Tables

B.1 An example of a batch file for running PMX ................................. 12
B.2 PMX source for excerpt of F.J. Haydn quartet ............................ 16
B.3 Meter options for mtrdepl = 0 .............................................. 19
B.4 Example of preamble parameters for the F.J. Haydn quartet in Fig. B.1, p. 15 22
B.5 Use of Note Parameters as shown in Fig. B.10 ............................ 29
B.6 Ornaments ........................................................................... 39
B.7 Parameters of the bar symbol R .............................................. 58
B.8 Symbols beginning with an A (global options) .............................. 71
B.9 inlinesample.tex, as produced from inlinesample.pm defects ........ 82

C.1 Mnemonics for instruments acceptable in PMX ............................ 90
C.2 The General MIDI Instrument Specification ................................ 92

D.1 Numerical limits of PMX variables (soft limits) ............................ 97
D.2 Numerical limits of PMX variables (hard limits) ............................ 97
List of Figures

A.1 W.A. Mozart, Sonata K545, bars 1–2

B.1 F.J. Haydn, quartet Op.76, no.2, bars 1–4
B.2 C.C. Noack, sonata diabolica, first movement, bars 12–17
B.3 I. Stravinsky, agon, first movement, bars 1–2
B.4 Result of meter options for mtrdenp = 0
B.5 Examples of pickups in 4/4
B.6 Notation of the clefs
B.7 C. Debussy, Pelléas et Mélisande (excerpt)
B.8 PMX notation for pitch (second digit of note symbol)
B.9 Relative Octave Notations
B.10 Use of Note Parameters
B.11 Rests
B.12 Xtuplets
B.13 Xtuplets with Sicherman brackets
B.14 Chords
B.15 Grace Notes
B.16 Ornaments
B.17 Beams
B.18 An extraordinary staff-jumping beam
B.19 Three realizations of simple slurs
B.20 Three realizations of exotic slurs
B.21 E. Bloch, Waves (Poems of the Sea I), bars 25,26
B.22 A dotted slur
B.23 Placing the slur ending with a staff-jumping beam
B.24 Shape variations in type K slurs
B.25 Vertical tweaks of slurs and ties
B.26 Dynamic marks
B.27 A clef change
B.28 A clef change in a staff with 2 voices
B.29 Figured bass in C. Monteverdi, L’Incoronazione di Poppea, aria “Pur ti miro”
B.30 Single bars, double bars, repeats
B.31 Volte
B.32 Some key changes
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.33</td>
<td>Some transpositions of a d minor scale</td>
<td>63</td>
</tr>
<tr>
<td>B.34</td>
<td>R. Wagner, Tristan und Isolde, beginning of third act</td>
<td>64</td>
</tr>
<tr>
<td>B.35</td>
<td>Titles, and text above and below a system</td>
<td>67</td>
</tr>
<tr>
<td>B.36</td>
<td>“Frère Jacques” (usage of macros)</td>
<td>78</td>
</tr>
<tr>
<td>B.37</td>
<td>A sample for the use of inline \TeX</td>
<td>81</td>
</tr>
<tr>
<td>C.1</td>
<td>The “baroque default”, and overriding it with MIDI only accidentals</td>
<td>93</td>
</tr>
<tr>
<td>C.2</td>
<td>Further use of MIDI only accidentals</td>
<td>93</td>
</tr>
<tr>
<td>E.1</td>
<td>J.Chr. Bach, Quartet in B Major (beginning of cello voice)</td>
<td>105</td>
</tr>
<tr>
<td>E.2</td>
<td>A. Bruckner, Locus iste</td>
<td>108</td>
</tr>
<tr>
<td>G.1</td>
<td>G. Dufay, Kyrie (generated by PMX)</td>
<td>113</td>
</tr>
<tr>
<td>G.2</td>
<td>A. Vivaldi, Mundi Rector (generated by M-Tx/PMX)</td>
<td>117</td>
</tr>
<tr>
<td>G.3</td>
<td>G. Caccini, Amor l’ali m’impenna (generated by M-Tx/PMX)</td>
<td>127</td>
</tr>
</tbody>
</table>
Chapter A

Introduction

How to use this tutorial

The purpose and function of this tutorial is, actually, twofold: for the novice with PMX, it should be a readable guide to finding out what PMX does (and what it doesn’t), how it is sensibly used, and what to do when you run into difficulties. For the seasoned user, on the other hand, it should serve as an extended PMX manual,

1. giving a reasonably accurate account of all the available features in an order which should make it not too tedious to find an answer to whatever question one might have,

2. indicating some of the more common uses of straight MusiXTeX commands to do things not available in PMX directly.

As a consequence of this twofold purpose, the two types of readers should have two very different approaches:

The novice should, after reading Section A 1, turn right to Section B 2 and B 4.1. After that you should get your fingers dirty as quickly as possible: keep your first own PMX file my_opus.pmx as simple as possible, and proudly produce your first print, following the instructions given in Section B 1. From thereon follow your interests!

The seasoned user will usually need the tutorial mainly to find out the exact usage of some commands he unforgiveably forgot about, retrieve some rarely used procedure, or find some clever gimmick that he never was aware of. Those people will first scan the whole tutorial quickly in a matter of minutes to see what’s there and gloat about what’s missing, and later on rely on the index – or look into authoritative source, the PMX manual by Don Simons.

Of course, once you have gotten acquainted well enough with PMX, all you’ll ever need is Don Simons’ “Quick Reference Table” 1. That summarizes the PMX symbols with all their

---

1 The Quick Reference Table is found in the software section of the Werner Icking Music Archive as ref260.pdf.
options and always reflects the latest version. Errors found in the Quick Reference Table should be reported to Don Simons directly, whereas all complaints (outright errors, awkward formulations, missed subtleties) should be sent solely to the author.

This tutorial refers to **PMX 2.613**, as of June 2011. For details on the differences with previous versions consult the list of changes of PMX in the software section of the WIMA ("Werner Icking Music Archive")².

### A 1 \TeX, MusiX\TeX, PMX, M-Tx

MusiX\TeX is undoubtedly one of the best programs for typesetting musical scores: it produces ready-to-print output in PostScript and PDF format, it is stable, is continually updated, and is in the public domain and thus totally free.

Nevertheless its use seems to be limited, with few exceptions, to musicians coming from the world of science. MusiX\TeX does not ‘look and feel’ very intuitive, and it sometimes requires a familiarity with terms that may seem elementary to software programmers but are tough going for straight musicians. Furthermore, it is not WYSIWYG software (What You See Is What You Get). This means that while writing the music you cannot immediately see the score as it will look when printed, because you do not write the music graphically as you do by hand, but rather in terms of a list of symbols that represent it. The system is not interactive; rather, the process of producing a musical score consists of several steps:

1. One writes the symbols in a special version of the \TeX language to a normal text file, using any text editor (such as Emacs, Edit or Notepad),

2. this text file is compiled with \TeX to produce a .dvi output file. This .dvi file is machine readable and usually can be previewed on the screen,

3. another program, e.g. dvips, produces a PostScript file from the .dvi file, which again can be previewed on the screen or sent directly to a suitable printer.

This symbolic and non-graphical nature of the program may discourage many people, but it is in many ways an advantage. Being fully aware of the symbolic character of the typesetting process while working on your input lets you concentrate much better on the final typographic output and keeps you from being distracted by the temptation to play games with interactivity.

MusiX\TeX is a collection of ‘macro’s that permits producing musical scores with the help of \TeX, the prestigious electronic typesetting program written by Donald Knuth, which is mainly used in scientific typesetting. In providing \TeX, Knuth not only put it in the public domain (to be freely distributed), but stipulated that any application that uses \TeX should also be in the public domain and freely available, without charge or royalty.

Before \TeX can be used, it has to be properly installed on the computer system at hand. Installing \TeX from scratch is not an easy task; consult Chapter A 2 for the details.

²For details on the staff-wise transposition feature refer to B 5.5.2.
Even after everything is properly installed and running, coding a musical score in the MusiXTeX language still remains a tedious process. Fortunately, there exist two preprocessors, PMX and M-Tx, which dramatically simplify the input process; in fact they provide what must be among the simplest possible systems of electronically encoding musical scores.

The first two bars of Mozart’s piano sonata KV 545 illuminate the difference in coding with either MusiXTeX, PMX or M-Tx. The codes given below for the three systems of software all produce the (identical) output, shown in figure A.1:

\begin{center}
\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figureA1.png}
\caption{W.A. Mozart, Sonata K545, bars 1–2}
\end{figure}
\end{center}

MusiXTeX:
\begin{verbatim}
\input musixtex
\parindent10mm
\setname1{Piano}
\setstaffs12
generalmeter{\meterfrac44}
nobarnumbers
\startextract
\Notes\ibu0f0\qb0{cge}\tbu0\qb0g|\hl j\en
\Notes\ibu0f0\qb0{cge}\tbu0\qb0g|\ql l\sk\ql n\en
\bar
\Notes\ibu0f0\qb0{dgf}|\qlp i\en
\Notes\tbu0\qb0g|\ibbl1j3\qb1j|\tb11|\qb1k\en
\Notes\ibu0f0\qb0{cge}\tbu0\qb0g|\hl j\en
\endextract
\end
\end{verbatim}

PMX:
\begin{verbatim}
2 1 4 4 4 4 0 0
1 1 20 0.12
Piano
\tt
/% Bars 1-2
c8 g+ e g c- g+ e g | d g f g c- g+ e g Rb /
c2+ e4 g | bd4- c1 d c2 /
\end{verbatim}

M-Tx:
\begin{verbatim}
```tex
\input musixtex
\parindent10mm
\setname1{Piano}
\setstaffs12
generalmeter{\meterfrac44}
nobarnumbers
\startextract
\Notes\ibu0f0\qb0{cge}\tbu0\qb0g|\hl j\en
\Notes\ibu0f0\qb0{cge}\tbu0\qb0g|\ql l\sk\ql n\en
\bar
\Notes\ibu0f0\qb0{dgf}|\qlp i\en
\Notes\tbu0\qb0g|\ibbl1j3\qb1j|\tb11|\qb1k\en
\Notes\ibu0f0\qb0{cge}\tbu0\qb0g|\hl j\en
\endextract
\end
```
**CHAPTER A. INTRODUCTION**

Style: piano  
Piano: Voices MD MS; Clefs G G; Continuo  
Name: Piano  
Meter: 4/4

\%
\%\% w120m
\%\% w120m
c2+ e4 g | b4d- c1 d c2 |  
c8+ g+ e g c- g+ e g | d g f g c- g+ e g |

Not only is the MusiXTeX much longer, it undoubtedly is less intuitive and more complicated than the other two.

In the present tutorial we shall describe in detail the usage of PMX and, in a very cursory way, that of M-Tx.

In the remaining part of this introduction, we treat questions of installation of the entire system of MusiXTeX 1.15 and PMX. The whole installation procedure may seem a formidable task to non-experts. But don’t despair: your efforts will be rewarded by the best and simplest system of musical typesetting available. And, to paraphrase a famous \TeX error message: “If in serious trouble, ask a wizard for help”. Such wizards are literally at your fingertips; just sign up and post your problem to the mailing list at WIMA (the Werner Icking Music Archive), and someone is bound to be in the mood to answer.

A 2  Installation

Recently, there has been a number of major improvements in the basic MusiXTeX 1.15 software (increasing, e.g., the maximum number of instruments allowed), and subsequently PMX has also undergone major improvements. As a consequence, PMX now is really tuned to MusiXTeX 1.15: with PMX(2.6xx) you can make full use of most of the new features of MusiXTeX 1.15.

The installation instructions given here refer explicitly to installing MusiXTeX 1.15 and PMX under Windows XP; they assume that you have MiKTeX 4 installed (version 2.6 or higher).

For Unix, good instructions can be found in the software section of WIMA: http://icking-music-archive.org/software/musixtex/musixtex-for-unix.html.

Installation of MusiXTeX 115 and PMX 2.6

MusiXTeX

1.15 You can download the basic distribution for MiKTeX 115 from WIMA: icking-music-archive.org/software/musixtex/musixtex.zip.

---

3 An important restriction is: in PMX(2.6xx) the maximum number of staves is 24.
4 MiKTeX is a \TeX system for Microsoft Windows users.
TEX users whose installation is compliant with the TEX Directory Standard (TDS) might prefer the MusiXTEX 1.15 distribution available [here](http://icking-music-archive.org/software/musixtex/mxinsuse.pdf).

A comprehensive – and very readable – installation instruction manual for MusiXTEX 1.15 has recently been given by Andre van Ryckeghem and Don Simons:


on the WIMA software page, which you can download [here](http://icking-music-archive.org/software/musixtex/mxinsuse.pdf).

Even if you consider yourself an expert in installing high-level software, it is certainly wise to first read this manual carefully (it’s only 6 pages) before starting the installation.

One point, pointed out explicitly in the MusiXTEX 1.15 installation manual, is of particular importance:

> As Don Simons has pointed out, it is safer to load MusiXTEX 1.15 (and all its subfolders, of course) completely outside of where MiKTEX is loaded. But whether you do that or load MusiXTEX in a subfolder of MiKTEX (traditionally named ‘localmiktex’), you must enter the full name of the MusiXTEX main folder in the ‘roots’ list of the MiKTEX ‘settings’ menu; otherwise MiKTeX will not be able to access any of the MusiXTEX files.

Another way to installing MusiXTEX 1.15 – specifically geared to a TDS-compliant TEX data structure – has been described by Bob Tennent. The TDS-compliant MusiXTEX 1.15 directory is [here](http://icking-music-archive.org/software/musixtex/mxinsuse.pdf); the [Installation instruction](http://icking-music-archive.org/software/musixtex/mxinsuse.pdf) is terse, but clear.

After completion of the MusiXTEX 1.15 installation, to test it download sample.zip from WIMA to some temporary folder and unzip it, obtaining the 3 files sample.tex, sample.pmx, and runpmx.bat.

[The file sample.tex is for testing MusiXTEX 1.15; sample.pmx and runpmx.bat are for subsequently testing PMX],

Now run

```bash
etex sample.tex
```

in a command line.

Make sure that when using basic TEX you always run `etex.exe` (‘extended TEX’) and not `tex.exe`!! Although MiKTeX and I3TeX now use extended TEX by default, the 2 commands `tex.exe` and `etex.exe` still both exist and differ; PMX 2.6xx, will not run properly under ordinary `tex.exe`.
CHAPTER A. INTRODUCTION

Installation of PMX

Compared to MusiXTEX 1.15, installing PMX 2.6 is a cinch. All you need to do is

1. decide which particular version of the PMX 2.6 series you need: 2.603 is by now almost standard; the new version 2.613 has a very important new feature regarding transposing instruments.

2. download pmx 2.613.zip (or 2.603) and unzip it,

3. copy pmxab.exe and scor2prt.exe to ...\localmixtex\miktex\bin,

4. copy pmx.tex to ...\localmixtex\tex,

5. run the MiKTeX FNDB (‘File Name Data Base’). 5

To test the installation, run runpmx sample.pmx in a command line.

A 3 Authors of the Software

MusiXTEX: Daniel Taupin

MusiXTEX evolved from MusicTEX by Daniel Taupin, a physicist in the Laboratoire de Physique des Solides in Orsay, Ross Mitchell (CSIRO Division of Atmospheric Research, Private Bag No.1, Mordialloc, Victoria 3195, Australia) and Andreas Egler (Ruhr–Uni–Bochum, Ursulastr. 32, D-44793 Bochum).

MusiXTEX had been maintained actively and conscientiously by Daniel Taupin until his untimely death in 2001 — he died accidentally while climbing in the Alps. He is remembered with a Memorial on the web page of the Werner Icking Music Archive.

After Taupins death, MusiXTEX seemed quite stable and could thus, for all practical purposes, be considered ‘frozen’ for a long time.

However, nothing as lively as MusiXTEX can really remain literally frozen forever. A number of improvements have been continuously added by a group of MusiXTEXperts, and, finally, Hiroaki Morimoto has kindly provided a new version of MusiXTEX which corrects known bugs. Then a new and strongly enhanced version of the MusiXTEX manual was prepared by a group led by Don Simons, which has now been officially released: MusiXTEX 1.15. It is located at the software section of the Werner Icking Music Archive (dubbed ‘WIMA’).

Finally, here is a list of the authors of essential parts of PMX and related software:

5 As was pointed out in the warning note above, it is advisable to load MusiXTEX 1.15 outside of where MiKTeX is loaded, and you can choose the name of the top folder for MusiXTeX 1.15 freely. The conventional name localmiktex is used here for simplicity only; it is not to be understood literally.
A 4. THE WERNER ICKING MUSIC ARCHIVE

Werner Icking (June 25, 1943 – February 8, 2001) was the founder of the GMD Music Archive, which contained downloadable scores, various MusiXTeX programs, a discussion list, and various other musical resources in the public domain. His contributions to the development and promulgation of this software were peerless. His editions of music (in particular the edition of the complete ‘Urtext’ of J.S. Bach’s pieces for violin and violoncello solo) are examples of sophisticated technique, artistic sensibility and philological meticulousness. Everybody, whether expert programmer or complete novice, would always find valid and unselfish advice from him, which invariably ended with the motto “hope this hilft”. In time a small, but regular group of faithfuls gathered around this site, from all over the world, who never had met Werner in person, but nevertheless came to see a friend in him, mainly through frequent email exchanges.

Werner died suddenly on February 8, 2001, while bicycling home from work, as was his regular habit. That very day he had performed maintenance work for the site and replied to messages on its discussion list, of which he had been the main administrator. The community of users of the archive decided not to disband, but to continue Werner’s work. Starting with the contents of the GMD Archive, a new site was created, and dedicated to Werner’s memory. It’s official name is the Werner Icking Music Archive.

After some difficult times, it is now hosted by PALDAM IT; its URL is http://icking-music-archive.org. Its chief administrator is Christian Mondrup, who more than any other provided reasoned guidance and continuity during the difficult transition.

The archive houses four main categories of information:

Archive of sheet music: Contains hundreds of royalty-free scores in PDF, many with their source code in MusiXTeX, PMX, or M-Tx. It is maintained by Christian Mondrup, who is also the main administrator of the Werner Icking Music Archive
E-mail: reccmo@daimi.au.dk.
MusiXTeX and related software: Contains MusiXTeX and related software, as well as user manuals and various add-ons. It is maintained by Don Simons
E-mail: dsimons@roadrunner.com.

\TeX-music list: This discussion list is invaluable to novices, advanced users and developers. Themes focus on technical problems of typesetting using MusiXTeX and related software, but often drift into other musical subjects of historic or esthetic interest. It is not moderated, but you need to subscribe in order to participate. For information on how to subscribe, visit http://icking-music-archive.org/mailman/listinfo/tex-music. Its administrator is Maurizio Codogno (E-mail qve-a159@myamail.com).

Links to other sources of information regarding music: There are links to search facilities about composers/works/editions, a multi-lingual glossary of musical terms, and some lists of composers’ works. This is maintained by Jean-Pierre Coulon:
E-mail: coulon@obs-nice.fr
Chapter B

A PMX Tutorial

Conventions for this tutorial

“Hey, this is boring stuff, but if you take a minute to understand the typographic conventions and a little jargon, it may avoid some confusion down the road.”

Typographical conventions:

- The **typewriter** typeface always indicates verbatim text *exactly as you would input it to the computer*. This includes file names, MusiXT\TeX tokens, and PMX commands, e.g., barsant.pmx, \internote, c44.

- **Bold** typeface is used for two purposes: first, for program names (e.g. pmxab), and second, when applied to a single letter within a normal word, to emphasize the mnemonics of a PMX command (e.g. lr signifies a “left-right repeat”).

- When used in commands, **italics** represent input variables for which the user would substitute the appropriate actual value. To make this notation quite clear – and distinct from the general use of italics for emphasis –, the variable will be surrounded by square brackets in such cases (e.g. L[n]P[m]M), but the brackets are not to be included with the verbatim text.

Musical terms:
The language of music is Italian, and so the radical solution of terminological problems would be to use the Italian names throughout. This would, however, defeat the whole purpose of this tutorial, so some compromises need to be made, and some terms, which may be used differently by different people, need to be clearly defined. So in this tutorial

---

†Quoted literally from the original introduction to PMX by its author, Don Simons.
• a *staff* (plural *staves*) means the set of 5 horizontal lines in which music is usually notated, aptly called a *pentagramma* in Italian.1

• a *system* means a coherent set of staves to be played simultaneously.

• a *voice* means a line of music that is a musical entity of its own, such as a melody. In polyphonic music, there is often more than one voice to a staff, e.g. a 4-voice Bach choral is often written with the 2 top voices (soprano, alto) in one staff, the bottom ones (tenor, basso) in the other 2.

**technical terms:**

• a *system* means a coherent set of staves to be played simultaneously.

• a *block* is the series of PMX commands WITHIN ONE SYSTEM, formally ended be a / (slash), which then must be the LAST CHARACTER on this input line 3.

### B 1 Running PMX

As stated in the introduction (cf. Section A), **PMX** is a *preprocessor* to MusiXTeX. That means that with **PMX** the sequence of steps from the input to the printed paper is one step longer:

0. The user writes *the symbols of the PMX language* to a normal text file, using any text editor. The name of this file, (say, *my_opus.pmx*) must have the suffix *pmx*.

1. the file *my_opus.pmx* is run through the **PMX** processor, **pmxab**. This produces the output file *my_opus.tex*.

   **Note that in PMX (like in Unix, but unlike Windows) spaces within a file name are not allowed!** If in the above example, the input file name were *my_opus.pmx* instead of *my_opus.pmx*, an unpredictable error, with confusing error messages, would occur.

2. the file *my_opus.tex* is then processed with **TeX** to produce a *.dvi* output file 4. This *.dvi* file is machine readable and (usually) can be previewed on the screen.

---

1 Note that one-line staves, used for percussion instruments (drums, triangles etc.), are presently not implemented in **PMX**.

2 **PMX** allows 1 or 2 voices in one staff.

3 This often precludes even **TeX** comments (started with a %) on the same line, after the slash! **PMX** does not check for compliance with this rule, but nevertheless it is good practice to adhere to it. Otherwise you are liable to encounter weird errors which may be difficult to trace.

4 This step is a bit more complicated than it sounds because it actually is a 3 pass system: it consists of (i) running **TeX**, (ii) running *musixflx*, and (iii) running **TeX** again (details on the rationale behind this are described in the MusiXTeX manual by Daniel Taupin).
3. another program, e.g. `dvips`, produces a PostScript file from the `.dvi` file, which again can be previewed on the screen, or sent directly to a suitable printer.

Usually this whole process is automated to some degree by the use of a so-called ‘script’ or ‘batch’ file, or with the help of some more elaborate system of interconnecting software (most of these tuned to the needs of the general TeX community).

There are too many variants of this kind of supporting software – public domain or commercial – to discuss here, and personal preferences or dispositions are too varied. If you are at a loss on what to use, send a message to the TeX-music list (cf. p.7), and you can trust that you get all the individual help you need.
Table B.1: An example of a batch file for running PMX
To illustrate the above process, however, an example for a typical ‘batch’ file, for a standard Windows system, is given in the above table (B.1). This assumes that you have installed

1. a MiKTeX system,
2. a MusiXTeX system,
3. a dvips system,

all installed in the standard way, and that you have produced the PMX input file my_opus.pmx, using your favorite ASCII text editor.

You would then run

\texttt{runpmx my\_opus}

on a command line, and would see a Ghostview window opening, from which you could print your opus (or do with it whatever you like).

You perhaps will want to write your own batch program\footnote{The runpmx.bat given in Table B.1 is identical with the runpmx.bat used in the installation guide (Section A 2).}

When adapting this to your needs or writing your own batch file altogether, you should make sure that you use \texttt{etex.exe} in the 3 calls of \TeX. MusiXTeX-T115 and PMX texttt2.6xx both require e\TeX (extended \TeX)! , to suit your personal requirements. Some further remarks on how pmxab operates may prove helpful for that.

First, whenever pmxab terminates due to a syntax error, the exit code is set 1 (0 when there are no errors). There are various ways of detecting this with batch commands, and then acting accordingly.

Second, pmxab always writes a file pmxaerr.dat containing a single number: 0 if it exited normally, otherwise the line number in the .pmx file where the syntax error was. With advanced batch programming techniques, this file can be opened and read, and if there was an input error, a text editor can be opened and the input point placed on the line with the error.

There have been several requests to allow PMX to keep running even after it detects an input error. This has not been done because in most cases, any error messages after the first one would be meaningless, or worse, uncorrected errors could cause crashes. In any event, all the output from pmxab will be stored in the log file \texttt{[filename].pml}.

B 1.1 Concatenating several files

Sometimes one wants to split the work on a score into several smaller parts, not only when otherwise you would exceed some PMX numerical limit (cf. Section D 1), but simply for practical reasons. Logically, one should distinguish two separate cases:
1. Within one and the same PMX program, there might be parts that recur identically in several regions of the full code, and one would not want to retyping these over and over again (both for efficiency and for clarity reasons). In such cases, you would want to have a command similar to the \texttt{\input} of \TeX, \LaTeX{} and many other programming languages. In PMX, such an \texttt{\input} command has the form \texttt{AR[file name]}, where \textit{file name} is the name of the (ASCII) file you want to include, and \texttt{AR[file name]} will insert this file at exactly that point in your PMX program where the \texttt{AR[file name]} is.

A possible use for this procedure, i.e. using the \texttt{AR} command, could be to save typing when you want to include your favorite option defaults that in fact you use in almost every piece of music you write with PMX.

But a simple “cut & paste” is arguably a quite practical and more straightforward way of achieving this goal; moreover, if you want to be fancy about it, PMX has a much more sophisticated tool to offer for that purpose\footnote{In fact, there is yet another fancy tool, i.e. including a ‘global’ file with the specific name \texttt{pmx.mod}, useful for special purposes. For details on this, refer to the PMX manual.}, viz. a macro (cf. Sec. B 7).

2. Sometimes may have reason to make several completely independent PMX files, producing their output separately. In the end, you would want to concatenate these pieces again. This can be easily done:

Suppose you have coded the three movements of a piano sonata in three separate PMX files: \texttt{son1.pmx}, \texttt{son2.pmx} and \texttt{son3.pmx}. To create one single file from these, proceed as follows:

(a) Process the three files .\texttt{pmx} separately, obtaining the three files \texttt{son1.tex}, \texttt{son2.tex} and \texttt{son3.tex},

(b) Create a \TeX{} file \texttt{son.tex} consisting of the following lines:

\begin{verbatim}
\input musixtex
\startmuflex
\let\startmuflex\empty
\let\endmuflex\endmuflex
\let\endmuflex\empty
\input son1
\input son2
\input son3
\endmuflex
\bye
\end{verbatim}

(c) Process the file \texttt{son.tex}, in the usual three passes (\TeX{} $\Rightarrow$ musixflx $\Rightarrow$ \TeX{}).

The resulting file \texttt{son.dvi} then contains the full score of the entire sonata.
Alternatively, you can concatenate the files using \LaTeX{}, or more specifically, the \texttt{mtxlatex} package. For details on this see Section C 4.

**String Quartet op. 76, No.2**

![String Quartet Op. 76, No.2](image)

Figure B.1: F.J. Haydn, *quartet Op. 76, no. 2*, bars 1–4

## B 2 Preliminary Concepts

The PMX code of a musical score consists of two parts, the *preamble* and the *body*. Lines with a % in column 1 are comment lines (as in \LaTeX{}); they are disregarded by the program \footnote{Note, however, the additional conventions of the \texttt{scor2prt} program (see Chapter C 1, pp. 85).}.

The *preamble* contains the general specifications for a score, some musical (e.g., number of instruments, meter, key signature), and some typographical (e.g., number of pages, number of staves per page).

The coding for the music itself is given in the *body*, which in turn usually has a *header*. As an example, consider the first few bars of Haydn's *quartet Op. 76, no. 2*. The music shown in Fig. B.1 is obtained from the PMX source code given in Table 16.

### B 3 Preamble

The *preamble* consists of one or several lines of numbers, followed by lines with typographical data relevant for the whole score. The preamble ends with a line giving the name of the directory to which \texttt{pmxab} is to write its output \texttt{.tex} file.
F.J. Haydn, Quartet op.76, no.2, bars 1--4

**PREAMBLE:**
nstaves ninstr mtrnuml mtrdenl mtrnump mtrdenp
        4 4 4 4 0 6
npickup nkeys
        0  -1
npages nsystems musicsize fracindent
        1  1  16 .08

Violoncello
Viola
Violin II
Violin I
batt
. /

% **BODY:**
% **HEADER:**
Tc
F.J. Haydn
Tt
String Quartet op. 76, No.2
h
Allegro
Abep
w170m
% begin of music input -
% bar 1
d82 Df o. d+ o. d o. d o. r d o. d o. d o. d o. d o. d o. /
f83 Df o. f o. f o. f o. r f o. f o. f o. f o. f o. /
r8 a83 Df o. a o. a o. r a o. a o. a o. a o. /
a24 Df d- /

% bar 2
r8 d o. d o. d o. r e o. e o. e o. /
r8 g o. g o. g o. r g o. g o. g o. /
r8 b o. b o. b o. r cs o. c o. c o. /
e2 a- /

% bars 3-4
f8 s e f c s s d4 a | b2 s a4 s o. r Rb /
a2 t a4 t .c s- | d8 s e f d s e4 o. r /
d8 s c s d e s e s d c s a1+ s g s | f s g f e s d8 o. d o. cs4 r /
d8 s c s d e s g s f e s a | d4- zd+ cn1 s b a g s s a4 o. r /
% end of PMX source file

Table B.2: **PMX** source for excerpt of F.J. Haydn quartet

**B 3.1 Numerical input**

The first line(s) of the preamble must contain 12 numerical parameters, separated by one or more spaces (‘white space’). In **PMX**, as in **TeX**, a line feed is equivalent to a white space, so
the following 3 forms of the beginning numerical input are all equivalent:

```
4 4 4 4 0 6 0 -1
1 1 16 .08
— or —
4 4 4 4 0 6 0 -1 1 1 16 .08
— or —
```

For the purpose of this tutorial, the 12 s are named\(^8\).

\[
nstaves, ninstr,
\]

\[
mtrnuml, mtrdenl,
\]

\[
mtrnump, mtrdenp,
\]

\[
npickup,
\]

\[
nkeys,
\]

\[
npages, nsystems,
\]

\[
musicsize, fracindent
\]

Their significance is as follows:

- **nstaves**, an integer \(\leq 24\), is the total number of staves per system. Each staff may contain either one or two voices (lines of music). The number of voices in a staff may change as the piece progresses, but the total number of voices at any one time cannot exceed 24. So if, e.g., there are 24 staves, there can only be one voice per staff.

- **ninstr**, an integer \(\leq nstaves\), is the number of instruments. Each instrument has a unique name (cf. p. 21), and any instrument with more than one staff will have its staves joined with a curly bracket. Often there is only one staff per instrument, and **ninstr** = **nstaves** in that case. There are two ways to assign more than one staff to one or more instruments:

\(^8\)Their names are written in typewriter typeface here because in a way they are input and, as some of the examples show, used as such, albeit only in comment lines. But these names themselves never occur in PMX commands. (See, however, the footnote on p.98).
– if only the first, i.e. the lowest, instrument has more than one staff, such as in a score for piano and a one-staff solo instrument, simply make \texttt{ninstr} < \texttt{nstaves}, and any difference will show up in instrument 1, the bottom one in each system,
– When you need a more general assignment of staves to instruments, put a minus sign in front of \texttt{ninstr}, and follow \texttt{ninstr} with the number of staves in each instrument in succession, in sequence from the bottom one up (the same order as the instrument names, cf. p.21), separated by spaces. These numbers must add up to \texttt{nstaves}: in a way, they are simply a partition of the numerical parameter \texttt{ninstr}.

Example: The preamble of the first example (Fig. B.2) simply has \texttt{nstaves} = 3 and \texttt{noinst} = 2: while the Stravinsky excerpt of Fig. B.3 was obtained\(^9\) with \texttt{nstaves} = 9

\begin{verbatim}
Allegro

Oboe

Piano

Figure B.2: C.C. Noack, sonata diabolica, first movement, bars 12–17
\end{verbatim}

and \texttt{noinst} replaced by

\[-7\ 1\ 1\ 1\ 1\ 2\ 2\ 1\ .\]

This tells \texttt{PMX} that there are 7 instruments in all, with 1 staff each for the four strings, 2 each for the piano and the harp, and 1 for the trombone. Note that these numbers – 8 in all in this case – count as one parameter, i.e. the 12 numerical parameters of the preamble are made up of 19 numbers in this case:

\begin{verbatim}
9 1 1 1 2 2 1 : (nstaves)
-7 1 1 1 2 2 1 : (ninst)
4 8 : (mtrnuml, mtrdenl)
4 8 : (mtrnump, mtrdenp)
2 0 : (npickup, nkeys)
0 2 16 0.14 : (npages, nsystems, musicsize, fracindent)
\end{verbatim}

• The following 4 numbers serve to define the meter of the piece. The first pair, \texttt{mtrnuml} and \texttt{mtrdenl}, are the logical values which \texttt{PMX} uses to calculate the length of a bar: \texttt{mtrnuml} is the logical numerator of the meter, i.e. the number of beats per measure, \texttt{mtrdenl} the denominator.

\(^9\)In the \texttt{PMX} text with which Fig. B.3 was produced, an additional Musi\textsc{xt}\textsc{ex} command was used to gather the strings in a group in the usual way. This “inline use” of \textsc{TeX} commands will be explained in detail in Section B 8.
The second pair, \texttt{mtrnump} and \texttt{mtrdenp}, determines the appearance of the meter in the printed output, but has no effect on the internal timing analysis of \textsc{PMX}:

- if \texttt{mtrnump} > 0, then it and \texttt{mtrdenp} are printed literally as the numerator and denominator of the time signature. For example, \texttt{4 4 4 4} prints a standard 4 quarters meter,
- if \texttt{mtrnump} < 0, then the numerator actually used by \textsc{PMX} will still be the positive value of \texttt{mtrnump}, but the entire time signature will be printed with a vertical slash through it,
- if \texttt{mtrnump}=0, then \texttt{mtrdenp} determines the printed meter as given in Table B.3 and shown in Fig. B.4.

There are special rules for \texttt{n/16} and \texttt{n/1} time signatures (where the latter "1" normally means a semibreve or whole note). To get \texttt{n/1} time, use \texttt{mtrdenl} = 0 (zero) and \texttt{mtrdenp} = 1\textsuperscript{10}. For \texttt{n/16} time, it is \texttt{mtrdenl} = 1 and \texttt{mtrdenp} = 16. So the choice \texttt{[3 0 3 1]} for the 4 meter parameters, for example, will give a \texttt{3/1} time, both in the \textsc{PMX} internal computations and in the printed output, while a \texttt{5/16} time is indicated by \texttt{[5 1 5 16]}.

| 0 | no meter is printed ('blind' meter) |
| 1, 2, 3 or 4 | a single digit, between the 2nd and 4th lines of each staff |
| 5 | cut time (alla breve) |
| 6 | common time |
| 7 | numeral 3 with a vertical slash |

Table B.3: Meter options for \texttt{mtrdenp} = 0

- The 7\textsuperscript{th} parameter, \texttt{npickup}, is the number of beats in a pickup bar if one is present. If there is no pickup bar, set \texttt{npickup} = 0. \texttt{npickup} need not be an integer. Cf. Fig. B.5 for examples of pickup bars.

A pickup bar is the only bar that can have a different number of beats than the current value of \texttt{mtrnuml}. It must be followed with the first regular bar in the same block, i.e. the pickup note(s) must not be followed by a |, / or //.

We shall later describe a \textsc{PMX} command to change the meter, and explain how to use it for pickups to later sections or movements (cf. section B.5.3).

- \texttt{nkeys} is the key signature, positive integer for sharps, negative for flats.

\textsuperscript{10}To remember this rule, recall that the printed denominator is taken literally, while the logical denominator represents the same time value that is denoted when entering ordinary notes (see section B.4.1 below): there 0 stands for a whole note. The special rule for \texttt{n/16} time is due to programming convenience.
Figure B.3: I. Stravinsky, *agon*, first movement, bars 1–2
The last four numerical parameters concern the layout:

- If \texttt{npages} > 0, it is the number of pages, and \texttt{nsystems} is then the total number of systems in the entire piece. \texttt{PMX} will spread the entire piece horizontally over this number of systems, and vertically over \texttt{npages} pages. For proper vertical spacing there should be from about 9 to 16 staves per page. If you specify too many staves for a given number of pages, one or more staves may spill over onto an extra page, but you will not see this until you preview the \texttt{.dvi} file. Possible remedies are to increase \texttt{npages}, decrease \texttt{nsystems}, or use the \texttt{Ae} command, to be described later (cf. section B 6.1).

- If \texttt{npages} = 0, \texttt{nsystems} is interpreted as the average number of bars per system. This is useful while building up a file a little at a time. \texttt{PMX} will decide how many pages to use.

- \texttt{musicsize} is the height of a staff, in points. The only values allowed are: 16, 20, 24 or 29.

- Finally, \texttt{fracindent} is the indentation of the first system from the left margin, expressed as a decimal fraction of the total line width.

  Note that although logically \texttt{fracindent} could have any value between 0 and 1 , you should never use a value seriously over .5. It is not only esthetically rather unpleasing, but may cause some unexpected havoc for MusiXT\mbox{\TeX}.

In the next part of the preamble the names of the \texttt{ninstr} instruments are given, as you want them to appear within the indentation in the first system, one per line, \textit{starting with the bottom} instrument. If you don’t want instrument names to appear, you must still leave \texttt{ninstr} blank lines here.

Next, on a new line, comes a single string of \texttt{nstaves} letters or numbers for the clefs, again starting with the bottom staff. The choices are \textsuperscript{11}: \texttt{b, r, n, a, m, s, t, f}, which stand for

\footnotesize{\textsuperscript{11}There are several more exotic clefs, available only in straight MusiXT\mbox{\TeX} (cf. Section B 8 on “Inline \mbox{\TeX}”, and the MusiXT\mbox{\TeX} 1.15 manual, Section 2.21).}
Table B.4: Example of preamble parameters for the F.J. Haydn quartet in Fig. B.1, p. 15

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nstaves</td>
<td>4</td>
<td>(4 staves)</td>
</tr>
<tr>
<td>ninstr</td>
<td>4</td>
<td>(4 instruments)</td>
</tr>
<tr>
<td>mtrnuml</td>
<td>4</td>
<td>(4 beats ...</td>
</tr>
<tr>
<td>mtrdenl</td>
<td>4</td>
<td>...of quarters)</td>
</tr>
<tr>
<td>mtrnump</td>
<td>0</td>
<td>(option for ...</td>
</tr>
<tr>
<td>nmtrdenp</td>
<td>6</td>
<td>...common time)</td>
</tr>
<tr>
<td>npickup</td>
<td>0</td>
<td>(no pickup bar)</td>
</tr>
<tr>
<td>nkeys</td>
<td>-1</td>
<td>(key: d minor)</td>
</tr>
<tr>
<td>npages</td>
<td>1</td>
<td>(1 page)</td>
</tr>
<tr>
<td>nsystems</td>
<td>1</td>
<td>(1 system)</td>
</tr>
<tr>
<td>musicsize</td>
<td>16</td>
<td>(16 pt staff)</td>
</tr>
<tr>
<td>fracindent</td>
<td>.085</td>
<td>(indentation of 1st system)</td>
</tr>
</tbody>
</table>

bass, baritone, tenor, alto, mezzosoprano, soprano, treble, or french violin clef, respectively. Instead of these mnemonic abbreviations, you can use the digits 0–7, according to the notation 0 (treble), 1 (soprano), 2 (mezzosoprano), 3 (alto), 4 (tenor), 5 (bariton), 6 (bass) and 7 (french violin)\(^{12}\). These two clef codes are summarized in Fig. B.6.

Figure B.6: Notation of the clefs

The preamble ends with a line that contains the path name of the directory to which you want the files (the .tex file, in particular) to be written when pmxab processes your source text. Usually you will want that to be the current directory: in UNIX this is denoted by ./, in DOS (most versions, including Windows) it is .\. In any case, the path – and thus the whole preamble – must terminate with / or \.

The rest of the PMX input file is called the body. It usually begins with a number of global options, each on a line by itself, called the header (which can be empty).

Following the header, the input of the actual music begins. The basic unit is called a block, each one consisting of from 1 to 15 complete bars. The input data for all bars in a block are entered for each staff in turn, starting with the first (i.e. the lowest) staff. The sequence of staves (and thus of the instruments) corresponds to that in the preamble: the first is the bottom one of the system as it appears in the final score, and the last is the top one. The input for each staff ends with a / (slash). The data for one staff need not be contained on a single input line, but may spread over as many as needed, with any number of blank lines and

\(^{12}\)This rather unmemonic notation has been retained in PMX for compatibility with MusiXTeX.
comment lines between them\textsuperscript{13}, but after the end of one staff (i.e. after the slash), you \textit{must} start the coding for the next staff on a new input line. A \textit{block} ends with the slash of the last staff of the last instrument (the top staff in the score).

Though you can put up to 15 bars in one block, many users of \textbf{PMX} have the habit of systematically writing one bar per block only. Sometimes, however, for reasons of context, \textbf{PMX} requires a set of bars to be in one and the same block (cf. Section B 4.12). If you do put more that one bar in a block, it is advisable (although not required) to separate the bars with a $|$ (‘vertical stroke’). The main function of this is to provide visual separation in the input file, and to help isolate input errors: if you put a $|$ anywhere except at a bar end, the \textbf{PMX} processor will stop and show you where it detected the timing error. Otherwise (with a few minor exceptions\textsuperscript{14}) $|$ has no effect.

It is also good practice to separate the blocks with comment lines that state which bars are represented, as has been done in the input source for Fig. B.1 (p. 15).

If there is a pickup bar ($npickup > 0$), it must be in a block with the first full bar\textsuperscript{15}.

When there are 2 voices in a staff (e.g. for an organ or other polyphonic instrument, or in a choir or symphonic score), these are entered consecutively in the coding for that staff, again starting with the lower voice, and this voice is ended with a $//$ (‘double slash’). Here again, after the double slash, you \textit{must} start the second voice on a separate input line. The coding for this staff is then again concluded by a single slash.

The two bars from \textit{Pelléas et Mélisande} by Debussy in Fig. B.7 are an example. The bottom staff, that for the trumpet, has only one voice, while the top staff (for the flutes) has 2, thus giving a total of 3 voices.

The number of voices in a staff (1 or 2) is determined solely by whether the first sequence of symbols for that staff ends with / or $//$. Therefore the number of voices in any given staff can vary from block to block, but not within a block. In other words, if you need to change the number of voices in a staff, you must start a new block.

Finally, a general characteristic of \textbf{PMX} coding should be noted: all data comprises sequences of \textit{symbols}, each one containing one or more adjacent characters. These symbols – including the voice-terminating symbols / and $//$ – are \textit{always} separated from each other by at least one space or line feed. Whenever there are several characters strung together without spaces, they are considered \textit{one single} \textbf{PMX} symbol. This concept has already been illustrated in prior examples. The construction of the symbols will be explained in much more detail in the remainder of this chapter.

\textsuperscript{13}Remember that in \textbf{PMX}, as in \texttt{TeX}, a line feed is equivalent to a space, and a sequence of many spaces is equivalent to one space.

\textsuperscript{14}For such an exception cf. the note on p. 58.

\textsuperscript{15}If, for some reason, you do need to put the pickup in a separate block, set the initial logical meter to fit the pickup bar, then after the pickup bar do a blind meter change, as described in section B 5.3.
Figure B.7: C. Debussy, Pelléas et Melisande (excerpt)
B 4 Commands for the Individual Staves

B 4.1 Notes

The most important input item is of course a single note. Its two main features are its pitch and its duration.

The pitch is primarily – i.e. up to octaviation – indicated by the note name in lower case letters: c, d, e, f, g, a, b (do, re, mi, fa, sol, la, si). If the note is to have an accidental, the note name is followed without a space by s, f or n for a sharp, flat or natural; ss/ff for a double sharp/double flat.

The basic duration, i.e. the duration exclusive of a possible dot, is indicated by the first unsigned digit following the note name, again with no space: 9, 0, 2, 4, 8, 1, 3, 6 respectively for double-whole (breve), whole (semibreve), half, quarter, eighth (quaver), sixteenth (semiquaver), thirty-second, and sixty-fourth notes. For a consecutive sequence of notes of equal duration, this needs to be given explicitly only for the first note of the sequence. Subsequent, consecutive notes with no explicit duration will ‘inherit’ their basic durations from the most recent note with an explicit duration.

A dotted/doubly dotted note is indicated by the letter d/dd just about anywhere in the note symbol, after the note name, of course.

It is important to note that, in contrast to the basic duration, a dot is never inherited by a subsequent note.

The pitch of a note is, of course, not determined by the note name alone; the actual octave needs to be specified as well. For this, there are three possibilities:

explicit octaves:

A second unsigned digit indicates the octave to which the note belongs. For reference, octave 4 runs from middle C to the B above. The lowest note on an 88-key piano is the A in octave 0, while its highest note is the C in octave 8 (cf. also Fig. B.8).

inherited pitch:

For a note without any explicit octave or relative pitch indication, the octave is assigned by PMX such that the note is placed in the octave that makes it nearest to the most

---

16 The only restriction on the position of d within the note symbol occurs if the dot is to be shifted (cf. p. 27).
17 In contrast to the letters for accidentals and dots and other letters to be explained in Section B 4.4, the 2 digits for duration and pitch, if present, must always be given in this order. Specifically, if by omitting the duration digit you use the inherited duration, you can no longer specify the octave explicitly by an unsigned digit.
recent note *in the same voice*, i.e. the pitch is ‘inherited’. Thus for jumps of less than a fourth up or down, you only need to enter the note name to fully specify the pitch. This feature often lets you go for long stretches in a voice before needing to enter the octave. For jumps of a fifth or more, you need to specify the octave either explicitly as described above, or relatively as described below.

**relative pitch:**

A + or − (if not part of one of the signed numerical suboptions to be discussed later) indicates that a note is to be an octave higher or lower than it would otherwise be. Two consecutive +’s will raise the pitch two octaves, and so forth\(^\text{18}\).

As an example of all this, compare Fig. B.9 with the PMX notations given under the staff\(^\text{19}\).

\[\text{Figure B.9: Relative Octave Notations}\]

The first note of each line of music in a block must contain at minimum the note name and a basic duration value. It is good practice and can simplify editing if in addition an explicit octave is set there. However if it is not, PMX will make some assumptions: at the start of the first input block the pitch will be set as if there were a prior note of middle C. In later blocks PMX will use the obvious inheritance rules from the end of the prior block\(^\text{20}\).

### B 4.2 Dotted Notes

Inheritance of dotted notes is a little tricky. As was noted above, the dot itself is never inherited; you always have to use a d in the note symbol, even if the actual duration and octave are the same as the prior note. But the *basic* digit of duration need not be reentered if it hasn’t changed\(^\text{21}\). So for example, consecutive dotted half notes, each within a fourth of the previous one, could be most cleanly entered as cd24 ed gd ed, whereas cd24 e would represent a dotted half note followed by a plain half note, since the basic digit of duration was for a half note all along.

\(^\text{18}\)Two consecutive +’s can be abbreviated by one ++ .

\(^\text{19}\)The same result would be obtained with the full, absolute notation c8 d8 e8 c85 b84 a84 g84 g83 | c26 g23 | c04; but this is obviously much more verbose.

Explicit octave numbers can be combined with one or more + or −, although this is not recommended.

\(^\text{20}\)However, if the number of voices in a staff has changed from the prior block, it is safest to reset the octave at the start of a new block. Duration is never inherited across block ends, and thus must be reset at the start of each input block.

\(^\text{21}\)There is an exception to this: if you decide to indicate the octave with an explicit digit (‘absolute octave’), you must then enter the duration explicitly as well.
4.3 Stems

PMX usually determines the correct stem length and direction (up or down) of a note automatically. This can, however, be changed by the user, as needed:

- the stem direction is forced to go up with the letter u ('up') anywhere after the note name, or down with l ('lower'),

- the stem length of non-beamed notes can be shortened by adding the option S, followed by a decimal number between .5 and 4.0, representing the shortening in units of \internote. The shortening can be made “sticky”, i.e. applied to all following notes by adding a colon (:) after the number. Stickiness is terminated by S:.

4.4 Other Note Parameters

Here are some more options that can be appended to a note symbol (as usual without spaces):

Inhibited beaming (a)

Joining the note stems with beams is usually done automatically by PMX. Sometimes, however, you may want to inhibit this explicitly for a single note; this is achieved by the letter a (for alone).

Horizontal shift (e,r)

Sometimes you want to shift the horizontal position of a note, for example to avoid overlap with a note in another voice of the same staff. The letters e and r do that: e shifts the note to the left by its own width (a ‘notehead width’), while r shifts it to the right. An example of this can be seen in Fig. B.7.

Shift of accidentals (< / >)

Accidentals can be shifted too. One way is to enter + or - immediately after the accidental character, then an integer for the vertical shift in units of \internote, then another + or - , followed by the horizontal shift, given as a decimal fraction of a notehead width. If you use this method, you must enter both numbers. Alternatively – if you need only an horizontal shift –, you can simply use < or >, followed by the shift, in notehead widths. When shifting a sharp to avoid another sharp, a left shift of 0.85 is usually best. When shifting a flat to avoid a flat above it, a left shift of 0.3 is suggested.

Shift of the dot in dotted notes

The dot in dotted notes can be shifted analogously to the first method for accidentals, using signed numbers, except that the first (vertical) shift may be a decimal fraction.

Shorthand for dotted notes

There are two special shorthand notations for dotted rhythms. For normal dotted rhythms (3 : 1 ratio), if you include a period (.) in the note symbol, PMX will

\[\text{\footnotesize\textsuperscript{22}}\text{1 unit of } \backslash \text{internote} \text{ is roughly half the distance between staff lines.}\]

\[\text{\footnotesize\textsuperscript{23}}\text{For details of this cf. Section B 4.10.}\]
• assign a dot to the note just entered,

• terminate that note,

• prepare to receive the next note name without any space, and

• automatically assign a time value to the second note equal to one-third of the first one.

No time value may be entered for the second note, but octave and accidental parameters may. Ornaments and slurs (cf. Sections B 4.9 and B 4.11) following this symbol will apply to the second member. If you need to follow the main note with some modifying command, you can still use the shorthand (.) after that command and a space. The main advantage of this shorthand comes if you want to follow one dotted pair with another of the same rhythm; then you needn’t enter any explicit duration value for either member of the second pair. This is because after using the shorthand, the basic inherited duration value is set to that of the first note in the pair; remember that the basic duration does not include the dot!

For paired notes with 2:1 rhythmic ratios, the symbol , (comma) behaves similarly to the . (period) for 3:1 rhythms as just described.

If you use this shorthand for dotted pairs, either embedded in a single symbol or to start a separate symbol for the second member, then you cannot shift the dot position. If you do need to shift the dot, you should not use the shorthand notation.

For reference, all the letter-type note parameters are listed in Table B.5. Examples demonstrating the use of most of them are given in Fig. B.10.
Accidentals:

- s  — sharp
- f  — flat
- n  — natural
- ss — double sharp
- ff — double flat
- sc, ssc, fc, ffc, nc — cautionary accidental (accidental in parenthesis)
- si, fi, ni — MiDI accidental
  [does do not appear in the printed score — cf. Section C 2.2]

Dotted Notes:

- d  — single dot
- dd — double dot
- .  — shorthand for 3 : 1 rhythm
- ,  — shorthand for 2 : 1 rhythm

Stems:

- u  — force the stem up
- l  — force the stem down (lower)

Shifts of Position:

- e  — shift the notehead left by the notehead width
- r  — shift the notehead right by the notehead width
- <  — left shift of accidental
- >  — right shift of accidental
- +, - — general shift of accidental

Beam Inhibit:

- a  — alone (cf. Section B 4.10)

Xtuplets:

- x  — (xtuplet) (cf. Section B 4.6)

Table B.5: Use of Note Parameters as shown in Fig. B.10.

B 4.5  Rests

Rests are denoted by the letter r in place of a note name. Rests have duration digits just like notes, but obviously no second digit for giving an octave. As inheritance goes, an r behaves just like a note: it can inherit its duration from the prior note or rest, and notes as well as rests can inherit their duration from prior rests.

Any rest or sequence of rests that occupies a full bar will by default be horizontally centered in the bar.

There are a few special notations for rests:

- **rp** (‘rest pause’) denotes a full-bar rest (whole rest), regardless of what the meter signature for the bar may be,

- **rpo** (‘rest pause off-center’). The option o suppresses the centering of a pause,
• **b** (‘blank’) denotes a blank rest, i.e. one that *does not appear in print*. Logically, it has a duration that is determined, as usual, either by an explicit duration specifier or by inheritance. Blank rests are most commonly used when there are two voices in a staff, and one drops out for some part of the current input bar\(^{24}\).

• **rm** (‘rest multi-bar’) followed – without space! – by an integer \(n\) generates the multi-bar rest symbol with the number \(n\) above it, signifying a rest for \(n\) full bars\(^{25}\).

Note that that there can be *only one instance* of a multibar rest in a given input block!

The default vertical position of a rest depends on whether there are one or two voices in the staff. For one voice it is just the MusiXTeX default: approximately centered on the middle line. On the other hand, in the lower voice in a two-voice staff, the rest is lowered by 4\(\text{\textbackslash internote}\), while in the upper line it is raised by 2\(\text{\textbackslash internote}\). The default can be manually overridden by appending + or - and an integer for the offset (in \(\text{\textbackslash internotes}\)) from the middle line of the staff. Note that this means that, if in the case of 2 voices in a staff you want to have the rest centered on the middle line, you need to code it as \(r+0\) — see the last bar in Fig. B.11.

Some samples of rests are given in Fig. B.11.

---

\(^{24}\)Another interesting application is when you have two homophonic voices in one staff and, for simplicity’s sake you want to write only one rest for both voices (cf. the last 2 bars in Fig. B.11).

\(^{25}\)This is used in particular when the parts for individual instruments are generated from a full score, as is done automatically by `scor2prt`; cf. Section C 1).
B 4.6 Xtuplets

PMX insists very strictly on checking that the sum of durations of all notes or rests in a bar add up to the total required by the bar’s meter. Thus, it cannot deal easily with modern extensively polyrhythmic scores, percussion in particular. The only polyrhythmic feature built into PMX are the traditional xtuplets: doublets, triplets etc., together with their usual notation.

Xtuplets (duplets, triplets, etc.) can have from 2 to 24 notes or rests. By default all notes in an xtuplet have the same duration; but some can be dotted or have twice the basic duration (cf. p. 32). The notation is as follows:

1. The symbol for the first note of an xtuplet begins exactly like a note symbol, with the name of the first note in the xtuplet, or an r if the xtuplet starts with a rest, followed, as the case may be, by an accidental, a d or dd, a + or explicit octave, and an optional duration digit. However, this duration, whether given explicitly or inherited from a previous note or augmented by a dot, is not the duration of the first note, but represents the total duration of the whole xtuplet.

2. If the xtuplet is to be unbeamed, add an ‘a’ (‘alone’) right after the first note.

3. Next – with no space, as usual – comes x (for ‘xtuplet’), followed by a one- or two-digit integer, for the number of notes in the xtuplet. If the first note is to be dotted, add the usual d, if it is to have twice the basic duration, add a D or F (cf. page 32 for more detail on this). The only options allowed after this begin with the letter n and control the printed appearance of the xtuplet:

   - If n is omitted, the xtuplet is printed in the standard way, i.e. with the xtuplet 3 printed over (or under, as the stemming may dictate) for, say, a triplet. Note also that by default PMX prints a bracket only if the xtuplet notes are unbeamed; otherwise just the xtuplet number is printed.

26 But such problems occur occasionally even in classical music. Thus, a real challenge to any PMX expert is to code the (in)famous third movement of Mozart’s Oboe Quartet KV 370, where suddenly the oboe changes to an alla breve (4/4), while the strings continue on in a 6/8 meter!

27 The default maximum of 24 can be changed. Cf. D 1 .
• If \( n \) is followed by an *unsigned* integer, this integer is taken as the number to be printed instead of the natural (default) one.

• If \( n \) is followed by the letter \( f \) (flip), the xtuplet number is flipped vertically from its default position.

• The position of the xtuplet number can be adjusted in the usual way with one or two *signed decimal numbers* following \( n \): the first is a vertical shift in units of \texttt{\textbackslash internote}, the (optional) second a horizontal shift, in notehead widths.

• If \( n \) is given, but followed by a space, thus ending the first-note symbol, *no number* at all will be printed.

• For a *non-beamed xtuplet only* a suboption \( s \) can be added to the \( n \) option by which the slope of the bracket can be adjusted; i.e., you can use \texttt{ns[n]}, where \( n \) is an *signed* integer (except zero): a positive integer will lift the end of the bracket upwards, while a negative one will move it downwards.

Note that this suboption will operate in a more elegant way for Sicherman-type xtuplets\(^{28}\) only: when the Sichermann option AT is *not in effect*, the bracket number (the 3 in a triplet, e.g.) will *not* follow the slope adjustment and might have to be tweaked separately.

4. The second through last notes of the xtuplet are then each given by a separate note symbol, containing the meaningful subset of the parameters permitted for notes or rests:

   (a) the note name as the first character. This is the only required character. It can be an \texttt{r} (for a rest), except that the *last note of an xtuplet cannot be a rest*\(^{29}\),

   (b) an accidental,

   (c) an octave change (+ or -). The octave may also be given explicitly; this is the only digit allowed,

   (d) a \texttt{d} (dot). The next note after the dotted one is automatically shortened to half the normal value,

   (e) the character \texttt{D} in the note symbol for any note in an xtuplet doubles the duration of this note. As this accounts for two notes of the xtuplet, it will decrease the expected number of notes in the xtuplet by one. If used for the first note of an xtuplet, \texttt{D} goes *before* an optional \texttt{n} parameter,

   (f) the character \texttt{F} is the same as \texttt{D}, except that in the printed output the doubled note will appear dotted (a notation Bach sometimes used). As with \texttt{D}, an \texttt{F} for the first note goes *before* an optional \texttt{n}.

   (g) grace notes are allowed in xtuplets.

No explicit duration is allowed in symbols for the second through last members of the xtuplet, the duration having been determined by the first note.

\(^{28}\)For Sichermann xtuplets cf. the next page.

\(^{29}\)Cf. however, Sec. E 2.2
Beaming of xtuplets is done automatically. If it is to be inhibited, add the alone option to the first-note symbol (cf. Section B 4.4).

Fig. B.12 displays some examples of xtuplets, with the PMX coding given below the staff:

![Xtuplets](image)

Figure B.12: Xtuplets

As was remarked at the end of Section B 3, the individual symbols of PMX generally consist of one or several characters strung together without spaces. In that sense, each note of an xtuplet is considered to be a separate PMX symbol; so these are separated by spaces as usual.

A special, non-standard bracket can be printed for non-beamed xtuplets only with an addition to PMX designed by Col. G.L. Sicherman. This prints the xtuplet number not above or below the bracket, but in line with it, as shown in Fig. B.13. To invoke this, simply add the option AT as a separate PMX symbol anywhere in the body of the source file: this will switch from standard to Sicherman brackets for all xtuplets in the score.

![Xtuplets with Sicherman brackets](image)

Figure B.13: Xtuplets with Sicherman brackets

If you want to use this for specific xtuplets only or otherwise customize your use of the option, you will have to use inline \TeX\ commands, following instructions given in `tuplet.tex` as comments.
To use the Sichermann option, the file \texttt{tuplet.tex} (available from the software section of the WIMA) must be installed in your \TeX system!

\section*{B 4.7 Chords}

\textit{Chordal notes} are notes that share a common stem and have the same duration. They are coded in \texttt{PMX} as follows:

\begin{itemize}
  \item One of the notes in the chord is chosen to be the ‘first note’. The first note is written as usual and as described in Section B 4.1; all other chordal notes are denoted by separate symbols starting with \texttt{z}, followed by a note name. The note name can optionally be followed by an accidental and an octave indicator (+ or -), but no absolute octave is allowed. And no duration value is allowed either, as this is logically determined by the first note.

  Legally the first note can be any one of the chord notes; but wise use of this freedom of choice is definitely called for, since

  \begin{itemize}
    \item the first note sets the default stem direction,
    \item in a beam it can affect the height and slope of the beam,
    \item if the chord is beamed with other notes, the default height and slope of the beam will be determined by the ‘first note’ of the chord.
  \end{itemize}

  If a beam joining chords looks bad, you can often fix it by changing which note acts as the main one [of course there are other ways to fine-tune beam parameters, as will be described in detail in Section B 4.10]. For further details, consult Section B 4.7.1.

  If the main note is dotted, the dot is obviously ‘inherited’ for all other notes as well, so writing another \texttt{d} would be redundant. But if you want to shift the dot of a chordal note, you \textbf{must add} a \texttt{d}, followed by the shift parameters, as described on p.27.

  \item Normally \texttt{PMX} will automatically set notes in a chord so that collisions are avoided (in a second, e.g.). If you don’t like the result, you can shift the note by one notehead width to the right with \texttt{r}, or to the left with \texttt{e}.

  \item The position of accidentals will also be adjusted automatically by \texttt{PMX} so as to avoid collisions. But, as with single notes, you can shift accidentals of chord notes manually,\footnote{Note that if, in addition, you use a + or - to denote a relative octave (cf. p.26), this sign must come \textit{after} the shift parameter for the accidental.} using the parameters described on p.27.

  If you manually shift \textit{any} accidental in a chord, then automatic shifting will be disabled for \textit{all} the accidentals in a chord, unless you preface the shift parameter with \texttt{A} (e.g. \texttt{zcsA<.5}), in which case the manual shift will be added to that applied automatically by \texttt{PMX}.
\end{itemize}
Another option that affects accidental positioning in chords is \( Ao \), entered in the main note symbol of a chord. It will force the accidentals in that chord to be posted in the order they come in the source (starting with the main note), each one as far to the right as it will go without crashing into a notehead, stem, or another accidental.

The stem length and direction of a chord are controlled by the first note, but may be manually overridden with \( u \) (up) or \( l \) (lower) in the first note symbol.

As described above, every chord note has its own note symbol. This means, as always: no space within a note symbol, but at least one space between the note symbols of a chord.

\[ 4.7.1 \text{ Arpeggio} \]

An arpeggio is, of course, simply a special form of a chord, denoted by a vertical wavy line in front of the chord proper. To write that in \textit{PMX}, place the symbol \( ? \) after the symbols, with a space as usual, for both the first and last note of the chord.

When doing that, you should remain aware of the fact that the wavy line is generally understood as being played from the bottom up: lowest note first, highest last.

So far, so easy. There are, however, a few special cases to note:

1. In an arpeggio, the wavy line often turns out to be crowded to the chord itself; this happens in particular when some of the chord notes have accidentals. In such cases one would want to move the arpeggio line away from the chord itself a bit, i.e. to the left.

   This is easily achieved by adding a \(-\) and an (unsigned) decimal number, without spaces, to one of the \( ? \) designating the arpeggio — it doesn’t matter which one.
2. Sometimes the composer may want to have the arpeggio played in a more virtuose way (on string instruments, in particular): the first 2 – or even 3! – of the chord notes simultaneously, as a double (or triple) stop, then the other chord notes as a true arpeggio. This can be done easily in PMX: simply place the first ? not after the starting chord note, but later, after one of the z... notes.

In this case is is even more important to remember the standard meaning of the wavy line in musical language that was just pointed out! So in general, it is best to stick with the order of the chord notes “bottom to top” unless you really know what you are doing.

3. In piano music, it is quite common that an arpeggio is meant to go across the chords of both hands. This also is easily implemented in PMX: simply place the first ? at the start of the left-hand chord and the second one at the end of the right-hand chord — as common sense would suggest!

To see all this in examples, look at how the chords in Fig. B.14 were coded. When comparing this figure with its code it should become obvious that care is called for when producing chords with accidentals, and arpeggios in particular: some of the chords in bars 1 and 2 look ugly, to say the least, and the arpeggio line in bar 3 as well as the first one in bar 4 seem to be downright violations of the standards of good typesetting practice.

B 4.8 Grace notes

A grace note symbol starts with a G. It is entered in its natural order, i.e. before the main note symbol for a normal grace, behind the main note symbol for an after-grace. After G comes any combination of the following options:

- A single digit representing the number of notes in the grace. The default is 1.
- m (for multiplicity) and a digit, representing the number of flags or beams. The default is 1; 0 is allowed.
- s (for slur): this joins all notes of the grace to the main note. No s is needed in the main note symbol.
- x for a slash (only for single graces),
- l or u to force the direction of the stem(s) as desired.

Next comes the only required character, the first grace-note name. No time value must be entered, but, if needed, an octave (relative or explicit) or an accidental can be given as in a

---

31To my knowledge, this feature of PMX (available from Version 2.503 on) has not been explicitly documented in the official PMX documentation. I am very grateful to André van Ryckeghem for drawing our attention to this feature in one of his contributions to the WIMA “Tips & Tricks” WEB page.

32The maximum number of notes in a grace is not fixed, but depends on circumstances (cf. Section D 1); but up to 16 is usually safe.
normal note. Second and later notes must follow immediately in sequence, set apart by spaces, likewise without any time value, and without any intervening symbols.

‘After’-graces (graces coming after a main note) are entered similarly to normal graces, with the same parameters as above; but in addition to the other parameters either an A (for After) or W (for Way-after) is entered, before the grace note(s).

After-grace symbols associate the grace note(s) with the prior main note and are therefore entered after the main note symbol.

By default, PMX will place graces or after-graces immediately before or after the main note, way-afters as far to the right as possible before the next note or bar line. If either type of after-grace is slurred, the slur will start on the main note and end on the last one in the grace.

For a grace (not for an ‘after’-grace), you can, if necessary, add some space between the grace and the main note by inserting an X followed by an unsigned decimal in the grace symbol.

Examples for grace notes are displayed in Fig. B.15.

\[33\]

It doesn’t matter where you insert this shift symbol, as long as it makes syntactical sense; to state this more simply: the X[n] may not be inserted before a number and not after the leading note of the grace.

This added space for a grace is indeed essentially a hard space, so that its name, X, is quite appropriate. And, as with almost all such cases, it is wise to keep in mind that ‘hard spaces’ are what their name implies: a hard – or brutal, if you will – last resort if nothing else seems to help. Used unwisely, adding a hard space to a grace is liable to produce rather ugly results, as is exemplified in Fig. B.15.
B 4.9 Ornaments

Symbols for ornaments are entered after their associated note symbol, separated, as usual, by a space.\footnote{The ornament, although associated with a specific note, is considered a separate PMX symbol.}

The ornaments available in PMX are listed in Table B.6 and are illustrated in Fig. B.16.\footnote{In fact, the only difference between staccato and pizzicato is the vertical positioning of the dot.}

Figure B.15: Grace Notes

All except the staccato, tenuto, down fermata and parentheses will appear above the staff; staccato\footnote{The ornament, although associated with a specific note, is considered a separate PMX symbol.} and tenuto appear just above or below the note head, depending on the stem.
### Table B.6: Ornaments

<table>
<thead>
<tr>
<th>Staccato</th>
<th>o.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenuto</td>
<td>o_</td>
</tr>
<tr>
<td>Accent</td>
<td>o&gt;</td>
</tr>
<tr>
<td>Sforzando</td>
<td>o^-</td>
</tr>
<tr>
<td>Pizzicato</td>
<td>ou</td>
</tr>
<tr>
<td>Spiccato</td>
<td>op</td>
</tr>
<tr>
<td>×</td>
<td>ox</td>
</tr>
<tr>
<td>+</td>
<td>o+</td>
</tr>
<tr>
<td>Caesura†</td>
<td>oc</td>
</tr>
<tr>
<td>Breath†</td>
<td>ob</td>
</tr>
<tr>
<td>Upper fermata</td>
<td>of</td>
</tr>
<tr>
<td>Lower fermata</td>
<td>ofd</td>
</tr>
<tr>
<td>Shake</td>
<td>ot</td>
</tr>
<tr>
<td>Mordent</td>
<td>om</td>
</tr>
<tr>
<td>Mordent [right parenthesis before notehead]</td>
<td>o)</td>
</tr>
<tr>
<td>Appoggiatura [left parenthesis before notehead]</td>
<td>o(</td>
</tr>
<tr>
<td>Trill†</td>
<td>oTz</td>
</tr>
<tr>
<td>Implicit trill†</td>
<td>oTtz</td>
</tr>
<tr>
<td>Segno†</td>
<td>ogz</td>
</tr>
</tbody>
</table>

**Editorial marks:**

<table>
<thead>
<tr>
<th>Editorial accidentals</th>
<th>oes oef oen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dubious accidental</td>
<td>oes? oef? oen?</td>
</tr>
<tr>
<td>Dubious note</td>
<td>oe?</td>
</tr>
</tbody>
</table>

† See explanation in main text
direction, the parentheses at the level of the note head, of course.

The trill and segno symbol are special in that they may have additional optional characters. Either trill symbol may include a decimal number to specify the length of the wavy line indicating the duration of trilling, in \texttt{noteskips}. The default is 1. Thus \texttt{0} will be a \texttt{tr} without any wavy line, and \texttt{0.5} is a wavy line of 2 \texttt{noteskips} without any \texttt{tr} symbol starting the wavy line.

A segno may only be entered in the first (lowest) voice. It may be immediately followed by a positive or negative integer, which indicates a number of points that it will be offset horizontally; and it will appear above every staff of the system.

Once the ornament type has been specified, most ornaments can be raised or lowered from their default position by appending a signed integer to the symbol, giving the vertical offset in \texttt{internotes}. Caesura and breath may in addition have a signed number, giving the horizontal shift from default in notehead widths. These two ornaments also differ from the others in their default horizontal position, which is 0.5 \texttt{noteskip} past the note.

An ornament can be automatically repeated on a series of consecutive notes, provided the notes are all in the same input block. To activate this feature, terminate the first ornament symbol with :. Then every note in that voice will have the same ornament until a note is followed by the repeat terminator \texttt{c}:.

\section*{B 4.10 Beams}

For the most part, PMX automatically takes care of the details of defining beams: selecting which notes are beamed together, and setting the angle, direction, height, and \textit{multiplicity} (the number of bars along the top or bottom). However, one may define a \textit{forced} beam – which overrides PMX’s selection of which notes are beamed together – by surrounding the included notes with [ and ], being certain to separate these beam symbols: [ and ] and their options, from the included note symbols with spaces. Conversely, if PMX’s selection is to beam notes, and you want to inhibit this for a single note, add the option \texttt{a} to the note symbol (cf. Section B 4.4).

One may also wish to edit certain features of a beam even when PMX’s grouping decision would otherwise be acceptable; this is done by adding some parameters to [ and ] :

\begin{itemize}
\item \texttt{u}, \texttt{l} or \texttt{f} will override PMX’s selection of the direction of the beam:
\begin{itemize}
\item \texttt{u} makes the beam go above the noteheads (upper beam),
\item \texttt{l} makes the beam go below the noteheads (lower beam), while
\item \texttt{f} will flip the beam from whatever PMX would do automatically;
\end{itemize}
\item \texttt{h} forces the beam to be horizontal;
\item \texttt{m} followed by a single digit (1, 2, 3 or 4) forces the \textit{multiplicity} of the beam;
\end{itemize}
• j joins a beam grouping to a prior one started in another staff (cf. p. 43).

The beam symbol constructed so far can be followed, without space, by one, two or three consecutive integers, each preceded with + or -:

1. first is an adjustment of the starting height, which thus determines the stem length of the first beamed note. It is given in \_internotes and may range from -30 to 30,

2. the second is a slope adjustment (again in the range from -30 to 30),

3. The third is an additional adjustment to the starting height, given in units of the beam thickness. It may range from 1 to 3 only. It always acts to increases the stem length. This would only be used in rare cases, e.g. to align consecutive horizontal beams which have internal multiplicity changes. An example is shown in Fig. B.17, bar f.

Note that for technical reasons all three integers must always be given with a sign (even the last one, although it always is positive). Note also that, since the meaning of these numbers is interpreted by \_PMX in their sequence, if you need only the second or only the third number you must not omit the preceding number(s), but rather give explicit 0’s for them.

Here are some further possibilities regarding beaming:

• By default, xtuplets (cf. Section B 4.6) are set apart with their own beam. To beam an xtuplet together with other non-xtuplets, just include it with the other notes in a forced beam.

• Rests may also be included within forced beams, provided they are shorter than quarter rests, and of course that they come between the first and last notes under the beam.

• Sometimes one may wish to define beamed groupings with subgroups joined by a single beam. The symbol ] [ , standing alone between two note symbols inside a forced beam, causes the multiplicity to decrease to unity and immediately increase to its natural value for the next note. An example of this, as well as the beaming together with an xtuplet, can be seen in bar 2 of the trombone voice of Fig. B.3: the two doubly-beamed groups connected by a single beam are generated by [ c15 c c ][ c c cx3 c c ].

• Related to this is a single-slope beam group, which is the same as described in the previous item except that there is no connecting beam between the beamed subgroups. For this option, the beamed subgroups are separated by ]-[ standing alone between two notes inside the forced beam.

• If there are large jumps in pitch between notes in a beam within a single staff, as a matter of taste you may wish to start the beam for example as an upper one and end it as a lower. \_PMX will never do this automatically, but you can accomplish it by forcing the beam with appropriately modified up/down-ness, starting level, and slope. If you use this technique, there are two details to note:
1. if there are any intermediate multiplicity changes, they will only be handled properly if the initially specified up-down-ness is consistent with the vertical position of the intermediate notes involved.\footnote{This is due to a fundamental technical limitation of MusiXTeX.}

2. for proper appearance in crowded scores you may wish to insert hardspace or shifts as described in Section B 6.4.
• Beams cannot normally jump staves. But if that is desired, start the beam normally in one voice, and terminate the part of the beam in that voice with \(j\). Then resume the beam in the other voice on a neighboring staff with \(j\), ending that part with the normal \(j\). For staff-jumping beams, it’s OK to have just a single note inside one or both of the partial beams. Each voice must, however, still have the right number of beats, so you will probably need to adjust the durations with blank rests after the first members of the beam and before the second.

With staff-jumping beams, some adjustment of the beam height and slope will usually be required. Also, sometimes the up/down-ness of the ending section must be overridden; you will know that this is necessary if on your first try the ending is shifted horizontally from the desired position. These editing commands may sometimes produce unexpected results, and some iteration may be required to get exactly what you want.

An example of a staff-jumping beam is seen in Fig. B.18.

• If the option : (colon) is appended to the starting symbol \([\) for a forced beam, then henceforth beams of the same length will be force-beamed similarly in that line of music until the end of the input block. Within an input block, the automatic forced beaming can also be stopped explicitly by starting a new regular forced beam.

But remember that PMX is actually quite good at beaming notes in the manner a human musician would normally expect; so it will be necessary to use this facility only in the special cases when you want (or need) beaming different from what PMX provides on its own. In such cases, consider this beaming automatism as a kind of shorthand; as such, it will be really worth your while only when you want to repeat the same ‘non-orthodox’ beaming an appreciable number of times.

Regarding automatic forced beaming, you should also be aware of the following features:

1. As with regular beams, you can have notes of different durations beamed together in an automatic forced beam. The rule in that case is: in the beams following the initial one, the next sequence(s) of notes with the *same total duration time* as that of the notes beamed together initially will then be beamed together automatically. As usual, notes given without duration will ‘inherit’ their durations from the most recent note with an explicit duration. All this is best seen in the following example:

\[
\begin{align*}
\text{\(\text{\(\text{\(\frac{3}{4}\)}\)}\)} & \quad \text{\(\text{\(\text{\(\frac{3}{4}\)}\)}\)} \\
\text{\(\text{\(\text{\(\frac{3}{4}\)}\)}\)} & \quad \text{\(\text{\(\text{\(\frac{3}{4}\)}\)}\)} \\
\text{\(\text{\(\text{\(\frac{3}{4}\)}\)}\)} & \quad \text{\(\text{\(\text{\(\frac{3}{4}\)}\)}\)} \\
\text{\(\text{\(\text{\(\frac{3}{4}\)}\)}\)} & \quad \text{\(\text{\(\text{\(\frac{3}{4}\)}\)}\)}
\end{align*}
\]

which is coded as follows:
CHAPTER B. A PMX TUTORIAL

Note how the automatic beaming ends at the end of the first input block. Leaving out the slash at the end of the fourth line would create havoc, because a full note cannot be beamed. But no slash is needed between the code for the first three bars.

2. **PMX** will never produce beams across bar lines. However, a somewhat fortuitous exception of this general feature of **PMX** are staff-jumping beams, and a specific example of this can be seen in Fig. B.18. This is *not an intended feature*, and its intentional use is not encouraged.

A fortiori **PMX** does not allow automatic *forced-beaming* across bar lines. Any attempt to try that will always result in a **PMX** error stop. Thus, if your automatic forced-beam pattern would cause a beam to cross a bar line, you have to terminate the automatic beaming properly before the end of the bar, or **PMX** will stop with an error.

![Extraordinary staff-jumping beam](image)

Figure B.18: An extraordinary staff-jumping beam

Some further examples of beaming and their **PMX** coding are given in Fig. B.17.

### B 4.11 Slurs and Ties

**Pick a package**

By default **PMX** will use MusiXTeX’s built-in, font-based slurs and hairpins (crescendi and diminuendi, see Section B 4.12, p. 53 ff). To use these, you don’t have to install any more software. But they do have drawbacks: the fonts used by MusiXTeX for font-based slurs come
in fixed sizes and shapes and are therefore often not flexible enough in complicated situations. In contrast, slurs and hairpins produced with the help of the PostScript language overcome many of the shortcomings of their font-based counterparts.

There are two independent “third-party” packages that provide MusiXTPX 1.15 and PMX users with PostScript-based slurs, ties, and hairpins:

**Type K** – This package was written by Stanislav Kneifl. It is directly supported by PMX and will be the focus of any future PMX enhancements. Detailed instructions for its use are given in the corresponding paragraph below.

**Type M** – This package, by Hiroaki Morimoto, is somewhat more flexible when used directly in MusiXTPX 1.15, but it requires the Metapost package to be installed in the \TeX{} installation. These are not directly supported by PMX, but are advertised to be fully compatible with default font-based slurs of MusiXTPX 1.15. From PMX’s standpoint they are not different from font-based slurs, and the same options and features that apply to font-based slurs should apply as well to these.

Both PostScript slur packages provide excellent slurs, ties and crescendi. In fact, for simple slurs, you can hardly discern the differences between the three possibilities, as can be seen in Fig. B.19 — although type M slurs are generally somewhat flatter, indeed. On the other hand, for steep and long slurs, the difference is appreciable, as can be seen in the example of Fig. B.20.

![Figure B.19: Three realizations of simple slurs](image)

Here are some additional factors to consider when deciding which type of slurs and hairpins to use with PMX:

- True ties, which are shaped differently from slurs, are only available with Type K\(^\text{37}\). In all cases, the starting and ending positions are different for ties and slurs\(^\text{38}\).

\(^{37}\)In MusiXTPX 1.15, font-based true ties are actually available; but they were never incorporated in PMX, now being superseded by type K slurs.

\(^{38}\)The ends of an ordinary slur are centered horizontally above or below the notehead, while tie ends are shifted inboard and closer to the midheight of the notehead.
• Font-based hairpins cannot wrap over a line break, but the PostScript ones can.

• Font-based hairpins cannot be longer than 68 mm.

• With either postscript package, the resulting markings do not show up on the screen if you view the score with one of the commonly used DVI viewers such as XDVI or YAP. To see them, you have to first produce a PostScript file from the dvi file (using, for example, dvips), which you can then view on the screen and print on your printer using Ghostview or some other such program. But if you use Type 1 PostScript fonts, as was recommended in Section B 4.11, you will have all this software at hand anyway!

Detailed instructions for installing the Type K or Type M packages are given in Section B 4.11.

B 4.11.1 General slur usage

This subsection explains commands that apply to all types of slurs, followed by separate subsections describing operational features specific to either font-based or Type K slurs, and finally a subsection on the use of Type M slurs.

Here are the PMX symbols that normally define slurs or ties, except for slurs to or from grace notes:

• An opening parenthesis, ‘(’, begins a slur, and a closing parenthesis, ‘)’, ends it; similarly, an opening brace, ‘{’, begins a tie, and a closing brace, ‘}’, ends it. The opening

---

39 Recall that the slurring of grace notes is defined entirely within the grace symbol, as described in Section B 4.8.
parenthesis/brace is placed before the first note, and the closing parenthesis/brace after the last note of the slur or tie. As always, these symbols (including options, see below) are separated from notes and other symbols by spaces.

- Alternatively to parentheses, you can use the letter s (for slur) both to begin and end a slur; but these come after both the beginning and ending note. This is thus a toggle, turning a slur off if it’s on, and starting a new one otherwise.

For ties (instead of slurs), use the letter t (for tie), also a toggle.

Note that slurs or ties may end on a rest, but they may not start on one. The default ending height of the tie (or slur) in that case will be that of the note on which it started (in other words, it will be horizontal); but you can vary that explicitly in the usual way.

Often you need to open a slur or tie while another one is already open in the same voice, e.g., with tied chords, or if a second slur connects a subgroup of notes within a longer slur. In such cases, PMX needs to be told which slur is which. For this purpose, the slur opening character [ s, t, ( ] can optionally be followed by a single-character ID code [ 0-9, A-Z ] to identify that particular slur. To close that slur, insert the ID code right after the corresponding closing character [ s, t, ) ]. If an ID code is used, it must be the second character in the symbol. You cannot use any ID that is already in use on a slur or tie for a new slur or tie. See Fig. B.21 for an example.

\[\text{Figure B.21: E. Bloch, Waves (Poems of the Sea I), bars 25,26}\]

---

40Braces – and similarly the tie toggles described subsequently – , although legal with all types of slur, are really useful only with Type-K slurs, since only with these they will produce a true tie.

41If you happen to think that this would be musically meaningless, you are wrong. Just watch a really good pianist (or conductor, for that matter) ending a piece, say in pp. It ain’t over until the fat lady has really finished singing: until she is really completely done with taking her hands off the keyboard! In other words: she visibly (and audibly!!) plays the slur over the final rest.

42The example uses Type K slurs.
Normally, you can leave further details of slur appearance to PMX; the result will usually satisfy. If not, you can change some features manually by additional options which go right after the initial character and any ID code, as usual without a space:

- The default vertical position above or below the note heads can be overridden with
  - u (upper),
  - l (lower) or, equivalently, d (down).
- The starting or ending point can be shifted from its default by entering one or two explicitly signed numbers: the first, which must be an integer, gives the vertical shift, in \internotes; the second, which may be decimal, the horizontal offset, in notehead widths.
- A dotted slur is obtained by adding the option b (for broken) in the symbol that starts the slur (cf. Fig. B.22).

\[
\begin{align*}
\text{\texttt{( g4 c ) (b [ e8 d c b ] ) | (u a4 f ) g2 Rb /}}
\end{align*}
\]

Figure B.22: A dotted slur

Slurs for staff-jumping beams

Often – in piano music in particular – when you have a staff-jumping beam (cf. p. 43), you may want to join the two parts by a slur as well. PMX will, however, get very confused if you try to put the beginning slur symbol in one staff and the corresponding ending symbol in a different staff; both slur symbols must always be in the same staff. The solution is to put the ending slur symbol after the invisible rest; but do so carefully: even though the invisible rests are indeed invisible, PMX accounts for the horizontal space it would need for visible rests, and that determines the length of the slur. So if you want to avoid awkwardly adjusting the length of the slur by hand, you should subdivide the invisible rest appropriately so that one part of it will be at the note of the other staff where you want the slur to end. The vertical position of the slur end will have to be done manually. Fig. B.23 shows an example.

B 4.11.2 Invoking and using Type K slurs

To activate the Type K package, all you need to do is to add a line in the header of the PMX source after the preamble, containing the symbol Ap\(^{43}\).

For type K slurs, some optional parameters can be used in the slur symbol to change the shape of the slur:

\(^{43}\)For further details on the use of \(A\) see Section B 6.1.
4. COMMANDS FOR THE INDIVIDUAL STAVES

• an f (for flatter) will flatten the slur a bit, while
• h, H or HH (for higher) will increase the slur’s curvature by increasing degrees, thereby raising (or lowering) its middle.

These parameters can be used in either the starting or the ending slur symbol. They do not work with ties, i.e. with symbol t, and PMX will complain if you try that. Examples are given in Fig. B.24.

The Type K package contains TpX macros that activate or deactivate an automatic vertical adjustment of slurs or ties to avoid tangencies with staff lines. In PMX these adjustments are switched off by default, because they may alter the endpoint positions from what you would normally expect. However, if you wish to use them, they may be switched on or off globally at the start of any input block, and locally with options to individual slur or tie commands.

To activate these adjustments globally (or deactivate, if previously activated), at the beginning of any input block enter Ap+s (or Ap-s) for slurs, and Ap+t or (Ap-t) for ties. To activate them for one-time use only, simply include p+t or p+s as an option in the symbol for the affected slur or tie. No deactivation is then needed.

You can also sequentially increase (Ap+c) or decrease (Ap-c) the default curvature of slurs or ties (again at the beginning of any input block), so that the result is cumulative. But after the cumulative change has reached “HH” or “ff”, it will remain there, and nothing more will happen.
An example is given in Fig. B.25. The distinctions are fairly subtle; but some would see the adjusted markings as more pleasing since on close inspection they clearly avoid tangencies with staff lines.

![Figure B.25: Vertical tweaks of slurs and ties]

**Line-break tie options**

There are two options that only affect ties that span line breaks:

1. by default a full tie is drawn at the beginning of the second line in such cases. There is a global option, \texttt{Ap+t}, which tells \textsc{Pmx} to use half ties – which are flattened at their left-hand ends – on the second line, but only in cases where the second tie segment is shorter than 15 pt. This option may be cancelled with \texttt{Ap-h},

2. the second option – which should not be used together with the first one! – affects both the first and second segments of linebreak slurs and ties. It is the global option \texttt{Apl}. It alters two defaults to give what some might consider an improved appearance: first, it causes the first segment to be drawn as a normal tie, whereas by default it has a different shape than a normal tie; second, it moves the starting point of the second segment a bit to the left.

The \texttt{Apl} option has been enhanced since \textsc{Pmx} version 2.411; the main new feature is that it now works for slurs as well as for ties. Other enhancements are more subtle; read the description of these in the announcement text for version 2.411, given here:

```
“Apl in the preamble now activates special treatment of line-break slurs and ties (before, Apl only affected ties.).

“Specifically, it enables tweaking the ending position of the first segment (seg 1), the starting position of the second (seg 2), and the curvature of either, or both. [You must have musixps.tex available to TeX!] With Apl every slur/tie at a line break is automagically broken into two separate ones (no additional PMX
```
slur start or ending commands are required). Vertical/horizontal tweaks to the
start of the seg 1 and the end of seg 2 are handled as before. Vertical/horizontal
tweaks for the end of the seg 1 and start of seg 2 are entered as options in the
otherwise normal command that starts the slur/tie. The option for the end
of seg 1 starts with ”s” (for sever or split), then the usual one or two signed
numbers, then a second ”s” and one or two more signed numbers for the start
of seg 2.

"The usual curvature options h, H, HH, f, if included in the starting command
for a linebreak slur, will apply to seg 1, and if in the closing command, to seg 2.
Of course, if the slur/tie does not come at a linebreak, the special position
tweaks (those after the ”s” option) will all be ignored, and the curvature tweaks
on the closing note take precedence (as before)."

Without either of these options, the second segment sometimes becomes too
short to be clearly visible.

The global options mentioned (and in fact any of the other global options to be described
later) can be combined into a single symbol, e.g., Ap+sl.

B 4.11.3 Use of Type M Slurs and Ties

In contrast to the type K slurs, type M slurs are not explicitly supported by PMX; as a
consequence, you have to do some things manually. But it’s not difficult:

- The first step is to insert into the header of your PMX source file the following line
\input musixpss\relax .

- To process a PMX file with type M slurs, three extra steps are needed to produce a
printed musical score. Here is the full sequence of processing steps, for a file named
my_opus:

<table>
<thead>
<tr>
<th>command</th>
<th>resulting new file(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. text editor</td>
<td>my_opus.pmx</td>
</tr>
<tr>
<td>1. pmx my_opus my_opus</td>
<td>my_opus.tex</td>
</tr>
<tr>
<td>2a. tex my_opus my_opus.mx1, my_opus.slu</td>
<td></td>
</tr>
<tr>
<td>2b. musixflx my_opus my_opus.mx2</td>
<td></td>
</tr>
<tr>
<td>2c. tex my_opus my_opus.dvi</td>
<td></td>
</tr>
<tr>
<td>→ 3a. musixpss my_opus</td>
<td></td>
</tr>
<tr>
<td>→ 3b. mpost my_opus my_opus.mp, my_opus.1, ...</td>
<td></td>
</tr>
<tr>
<td>→ 3c. tex my_opus my_opus.dvi</td>
<td></td>
</tr>
<tr>
<td>4a. dvips my_opus my_opus.ps</td>
<td></td>
</tr>
<tr>
<td>4b. gsvier my_opus screen view and printed output</td>
<td></td>
</tr>
</tbody>
</table>


B 4.11.4 Special considerations for font-based slurs

You should study this section if you choose not to use Type K PostScript slurs.

The slur command t (cf. p.47) has been retained for backward compatibility with earlier PMX versions. As already mentioned, with font-based slurs it does not provide a true tie at all, and it doesn’t alter the positions of the endpoints either; in fact, it differs from s or ( only in the following ways:

- ID codes cannot be used with font-based t-slurs.
- If a t-slur starts or ends on the same note as an s-slur, the former will be moved away from the notehead to avoid a collision. This only works if neither slur has an ID code.

The shape of font-based slurs can be changed with options to the s or ) command as follows:

- at the slur termination only, you must first enter two signed numbers that define a position shift as described earlier. Enter a zero for either or both if you don’t want the shift,
- next, enter a signed, nonzero integer which specifies a vertical adjustment to the mid-height of the slur in \internotes,
- if you want to alter the starting or ending slope, enter a : (colon), then
  1. a signed integer for the alteration to the starting slope, then
  2. a signed integer for the alteration to the ending slope.

These numbers are passed directly as arguments of the MusiXTpX macros \midslur (if only one is given) or \curve (if there are three).

If you want to simulate a tie with font-based slurs – PMX does not provide true ties with font-based slurs! – include the option t in both the starting and ending command for the slur. This will draw a slur with altered endpoint positions.

B 4.12 Dynamic Marks

PMX has a very simple tool for inserting virtually any dynamic mark in a score. Its Dynamics symbol is a D, followed one of 3 types of parameters, optionally followed by position shifts:

1. A standard dynamic mark : this can be any one of the following: pppp, ppp, pp, p, mp, mf, f, ff, fff, ffff, sfz, fp.

2. A crescendo/diminuendo : < or >. These crescendo/decrescendo marks – aptly named “hairpins” – are toggles, i.e. the first D< starts the crescendo, and the next D< ends it; similarly for D>.

\footnote{For reasons of backward compatibility, it is legal to specify a tie with the symbol st or (t even when PostScript ties are activated. In that case, these symbols are fully equivalent to simply using t.}
3. **Arbitrary text**: "...", where ... stands for any text string, e.g. D"molto espressivo". The text will be set in italics unless another \TeX-style font specification is included.\(^{45}\)

All dynamics symbols go after the note to which they refer (so e.g. g Dpp will write a pianissimo g). Hairpin marks must be contained completely within the same input block.\(^{46}\)

The default position of any dynamic mark or text entered with the D symbol is just below the notehead, stem end, or bottom staff line, whichever of these is lowest. There are numerous context-sensitive automagic adjustments to the positions of all the dynamics symbols. If you don’t like the result you can add position shifts to the dynamics symbol: a signed integer for a vertical shift (in \internotes), then – optionally – another signed number for a horizontal shift (in notehead widths).

There can be a combination of dynamic marks at a single note.\(^{47}\) Since these are distinct PMX symbols, they must be separated by spaces and must come in the right order, e.g.

\[
\begin{align*}
\text{[some notes]} & \quad \text{D< [more notes]} & \quad \text{D< Dfff D> [more notes] D>}
\end{align*}
\]

If you aren’t using PostScript slurs, there are some restrictions on hairpins which are due to MusiX\TeX’s use of font-based hairpins: they cannot be longer than 68mm, they cannot wrap over a system break, and they must be horizontal. Finally, only certain specific lengths are available, so some horizontal position tweaking may be needed, especially when standard dynamic marks and hairpins are combined. These restrictions don’t apply when using one of the PostScript slur packages; PostScript hairpins will then be used without any further intervention.\(^{48}\) Fig. B.26 (a fragment from the Pathetique by Tschaikowsky) shows some examples.\(^{49}\)

### B 4.13 Clef Changes

As was noted in Section B 3, the clef for each instrument is given in the preamble (cf. p. \(21\)). A clef change in mid-stream, however – frequently occurring in piano music –, is signaled by a C followed by a single lower-case letter, as illustrated in Fig. B.27.

When there are two voices in a staff, the clef change command must be given in the first (lower) voice; if it is entered in the second voice, PMX will silently disregard the clef change completely. The clef change holds for all voices in a staff, of course.

A logical effect is worth mentioning: when, before the clef change, there is a note the duration of which extends beyond the point of the clef change, its clef does not change. Fig. B.28 shows an example of such a situation.

\(^{45}\)A practical example is D" \\ppff fz" (cf. Tips and Tricks).
\(^{46}\)If need be (e.g. for crescendi/decrescendi across bar lines), you can adjust the position and length of a crescendo/decrescendo by adding position shifts as described.
\(^{47}\)It doesn’t make sense, of course, to have 2 different standard dynamic marks on the same note, and PMX will complain if you try that.
\(^{48}\)For details on invoking PostScript slurs, ties and hairpins cf. Section B 4.11 ff.
\(^{49}\)Fig. B.26 was obtained using the PostScript slur package type K.
CHAPTER B. A PMX TUTORIAL

Figure B.26: Dynamic marks

Figure B.27: A clef change

B 4.14 Octaviation

When in a voice the pitch of a number of notes extends far beyond the staff, resulting in an excessive number of ledger lines, readability is usually improved by using an ‘octavation’ notation: an horizontal line above the staff, starting with an “8” or “8va”, indicates that the notes under this line are to be played an octave higher than printed, and similarly “8 bassa”
or “8va bassa” indicate an octave lower.

In the present version, PMX does not provide a notation for this, so you will have to take recourse to inline \LaTeX\ coding. For an example see Section E.

B 4.15 Figured bass (basso continuo)

PMX lets you introduce the basso continuo notation (‘figured bass’) very simply, as illustrated in the example of Fig. B.29, which is the basso continuo of an excerpt of the famous aria “Pur ti miro” that concludes the Monteverdi opera L’Incoronazione di Poppea. Here are the rules:

- Figure symbols are entered – as plain numbers\footnote{As of PMX version 2.520, the numbers usually needed in figured notation, namely 2, 4, 5, 6 and 9, are available in a different graphic design with perhaps a more baroque-like look. If you want to use these, add an ‘s’ after the PMX string of every figure which you want to appear in that form. Note that for this to work properly, you must have the special fonts for these numbers installed. You will find figbas.zip, which provides these fonts, in the WIMA software page. figbas.zip contains 2 files that can be of help with the installation: README and test.tex.} – after their associated note symbols, with a space, as usual. As of PMX version 2.517, they are allowed in two staves, the first (lowest) staff and any other one. Enter the characters as they would appear from top to bottom, e.g. \( 6 \over 4 \) [6 over 4], or \( 6 \over 4 \over 2 \) [6 over 4 over 2].

- Accidentals that modify a number must be entered before the number. Note that the characters that signify accidentals are different here than for notes: flats are written as ‘-’ (minus), sharps as ‘#’, and naturals as ‘n’. For example, sharp third is \( #3 \) (or simply ‘\#’, in the standard shorthand), six (over) flat five is \( 6-5 \), and sharp six (over) four is \( #64 \).

- For each staff with figures, PMX positions all the figures within each system below the staff with their tops at the same level. If you want to change this vertical position default, you have two possibilities:

  **to lower a figure symbol:** prefix the figure symbol – *no space!* – with one or several ‘\_’ (underscore). Each underscore will lower the figure symbol by 4 lengths of \internote

\( \)
to raise a figure symbol: append the figure symbol – again no space – with a ‘+’ (plus) followed by an integral number for the number of \internote lengths by which to raise the figure.

The two options can be combined to provide full control over the vertical position of the figure symbol.

- If you want a figure symbol to align horizontally in the second tier, insert a `_` (underscore) as placeholder for the top tier, before the one you want to go on the second tier (cf the sharp in bar 16 of Fig. B.29).

- Sometimes you may need to enter a figure when there’s no bass note sounding. To do this, precede the figure symbol with three additional characters, x\[n]\[m]. Here \(n\) is a single-digit repeat count, and \(m\) is a single-digit duration time value, i.e., \(0, 2, 4, 8, 1\) or \(3\). This will offset the figure from the associated note by the specified duration value. For example, if the lowest voice contained c\(\text{o3 x3465}\), there would be a whole-note c, and 3 quarter notes later a figure 65 below the staff (cf. Fig. B.29, bars 2, 4, 6, 8, 10 and 14).

- There is also a continuation symbol, viz. a 0 (zero) followed immediately by an unsigned number. This produces a horizontal line under the bass note, starting just to the left and extending to the right by the given number of \noteskip’s (cf. Fig. B.29, bars 9, 11, 16 and 17). The height and length of the line are set by the current note’s level and \noteskip respectively. These can be mixed in with other figures to produce vertical stacks. If another figure follows in the same symbol, use : as a separator. For example, a continuation line over a 3 would be coded as 01:3.

If there are figured bass commands in a PMX file, but you want them all to be ignored, then enter the symbol F in the header. This feature is most useful in the form %1F (cf. Sec. C 1, which makes a separate bass part with no figures, e.g. for a violoncello part.

Figured bass symbols will not be altered in any way under transposition by PMX (cf. Sect. B 5.5). There is no universal set of interpretations of figured bass symbols, so no automatic transposition is possible.

PMX does not admit 2-digit basso continuo figures. This is the notation of some old editions; if you want to use it, you will have to resort to inline \TeX coding (cf. Section B 8). There is an example for this in the Caccini aria in the Appendix (Sec. G 2.3).

### B 5 Commands That Affect All Voices

Unless indicated otherwise, the commands dealt with in this section affect all staves in a score. They must be entered only in the first (lowest) voice in the first (lowest) staff.

---

51 If \noteskip changes, or if an unfigured note drops below the starting level before the line ends, it is possible to trick PMX by entering separate 0\[n] symbols under each consecutive note: PMX will automagically join them together at the lower height (thanks to Werner Icking for this idea).

For an explanation of the special PMX coding in Bar 16 of Fig. B.29 see Section E.
B 5. COMMANDS THAT AFFECT ALL VOICES

Such commands will automatically be transferred from score to parts when separate parts are generated by `scor2prt` (cf. Section C 1).

B 5.1 Single bars, Double bars, Repeats etc.

In keeping with general typesetting practice, PMX will by default produce

- a single bar line at the beginning of each system after the first, unless there is only one staff per system,
Table B.7: Parameters of the bar symbol $R$

- a single bar line at the end of each bar except the last one in a movement or the entire score,
- the common ending bar line (thin-thick double bar line) as the last bar line of a movement or the entire score.

The user can, however, override these defaults – except the first one – by a bar symbol, viz. an $R$ followed by one of the parameters given in Table B.7. If you are going to make parts from your score (cf. Section C 1), you must place bar symbols either before the first note in an input block or after the last one; otherwise scor2prt may behave erratically. Using two bar symbols in succession doesn’t make sense and will produce unpredictable results.  

Here are further comments on some particular bar symbols:

- $R_b$ forces a single bar before a movement break, where the default would be a double bar. This can be useful, for example if you change the number of instruments, which PMX will allow only at a movement break, but you don’t want it to look like a movement break.

- $R_z$ will cause an invisible bar line at the end of the current system. It can be used together with blind meter changes (cf. Section B 5.3) when you want to split a bar across a system break.

- if a left-right repeat ($R_{lr}$) comes at a system break, PMX will automatically split it in two.

- $R_{dl}$ is the same as a left repeat ($R_l$) except at a system break: there the first system is ended with two thin lines, and the left repeat sign goes to the beginning of the next system.

- Finally, a general remark is in order regarding the $R$ symbols that designate repeats, double bars etc. ($R_l, R_r, R_{lr}, R_d, R_D, R_{dl}, R_b, R_z$). To avoid conflicts with scor2prt,

---

If you are using | marks at the end of bars, a repeat must always go before the |, or the next line must follow in the same input block. In other words: while $R_r | /$ will not result in an error, it will produce a thin bar line only.

There is a simple consequence: avoid using | altogether, at least in connection with $R$ symbols.
you should *always* place any `r` command before the first note in an input block. This is always possible except at the very end of the whole piece. Otherwise, there is always a following input block where the command can (and should) be placed.

Fig. B.30 gives some examples.

Figure B.30: Single bars, double bars, repeats

On some special occasions, you may want dashed or dotted barlines instead of the regular ones. For this and similar purposes, there is a very good and easy-to-use package by Rainer Dunker: *musixdbr.tex*, included in the MusiXTEX 1.15 distribution.

### B 5.2 Volta

The two versions of the ending of a repeated section of music, called ‘volte’ in musical parlance[^53], are denoted by a symbol starting with a `V` (for *Volta*) in PMX. PMX needs to know where each volta starts and ends, how the ends are shaped, and what text is to be included under each volta. A string parameter to the `V` is used to convey this information:

- to signal the *start* of the first volta, add a text string (usually just a 1) that doesn’t start with one of the characters `b`, `x`, or any literal spaces[^54]. PMX will print append a period to the text string.

---

[^53]: The term ‘volta’ is, of course, Italian, and ‘volte’ is the Italian plural of ‘volta’.

[^54]: Actually you can include a space by using the TeX space symbol `\` instead of a blank.
to signal the end of a volta and the start of the second volta, use a V with a text string that does start with either the character b or x. If b is present, a volta symbol ending with a vertical stroke will be printed (box), while x will print the ending without a vertical stroke (no box) \(55\).

Often composers will write out only the first ending of a repeated section of music, and simply continue play, dropping the first ending after the repeat. With PMX, the notation is analogous: simply end the first volta, with either Vb or Vx, without adding a second volta.

On the other hand, if one volta starts where another one ends, the symbols ending the first and starting the second volta can be combined into one, e.g. Vb2.

Other rules governing volte are:

- Volte should be placed at the beginning of an input block, in the first (lowest) voice of the first (lowest) staff, before the first note if it’s the start of a volta, or after the last note if it’s the end of a volta.

- If you will be making parts from the score using scor2prt, then to ensure that the volte are properly transferred to the parts, you must only include one volta symbol V in each input block, and it must come at the beginning of the block.

- If a score ends while a volta is still open, PMX will close it with a box.

Examples of volta usage can be seen in Fig. B.31.

---

Note that the characters b or x can appear anywhere in the text string and will not appear in print. \(55\)
B 5.3 Meter Changes

The meter of a piece can only be changed at the beginning of an input block, and thus naturally only at the beginning of a new bar, just before the beginning of the first (lowest) voice of the first (lowest) staff. As with the initial meter specified in the preamble, any new meter applies to all instruments and voices.

A meter change symbol starts with the letter \( m \). There are two different ways to complete the symbol:

- enter the 4 meter-defining numbers \( m\text{trnum}_1 \), \( m\text{trden}_1 \), \( m\text{trnump} \), \( m\text{trdenp} \) for the new meter, separated by slashes /.
- same as above, but enter the 4 numbers consecutively without spaces. In this case, it is necessary to distinguish between one single 2-digit number and two consecutive 1-digit numbers. For this – and for this case only – the following convention is adopted:
  - the number 1 is represented by the letter \( o \),
  - consecutive digits \( 10 \ldots 19 \) stand for exactly that: the 2-digit numbers 10 through 19.

Thus, 19 is the largest number that can be entered with this method.

Note that \( m\text{trden}_1=0 \) still represents a whole note (semibreve), as explained on p.19, and a ‘blind’ meter is given by \( m\text{trnump}=m\text{trdenp}=0 \), as explained in Table B.3.

The most common application of a ‘blind’ meter change occurs when a piece — often one starting with a pickup — has an incomplete final bar. In such cases place the last bar in an input block by itself, headed by a ‘blind’ meter change. For example, if the meter had been 4/4 and there was a quarter note pickup, leaving 3 beats in the last bar, the last bar might be coded \( m3\text{400} cd24 \) of /.

Another useful application is for the pickup bar in the second or later movements, following a PMX movement break symbol. Since there is no special provision for pickups in this case like there is at the beginning of a piece,

1. the pickup bar should be in a block by itself, starting with a meter change symbol with a logical meter representing the pickup bar and a printed meter as appropriate, and
2. the next bar should start a new block and begin with a blind meter change symbol with a logical meter representing the true meter.

B 5.4 Key Changes

A key change can be signalled at any time. It must be entered in the first voice, but will affect all voices. Use the command \( K+0 \), followed by the new key signature: positive integer for sharps, negative for flats (cf. Section B 3.1).

Some examples are given in Fig. B.32.

---

56 As was pointed out before (cf. p.31), this limits the use of PMX for modern polyrhythmic music. But you can always try to play with tricks . . .
57 For the meaning of these numbers cf. p.18.
58 Only one key change is allowed in each input block.
CHAPTER B. A PMX TUTORIAL

B 5.5 Transpositions

B 5.5.1 Transposition of an entire score

To transpose an entire score to a different key from the one in which the data have been entered, use the symbol $K$ (for Key), followed by two explicitly signed digits: (1) the distance to transpose by, in units of \texttt{\textbackslash internote} (i.e. the vertical offset on the staff), and (2) the new key signature. This symbol must be entered at the beginning of the piece, in the header (cf. p.22).

There is one case that requires special handling, viz., when transposing by a half step, but the vertical position doesn’t change, e.g. from D flat to D. In this case the shift parameter (the first parameter of $K$) is always $-0$ regardless of whether the transposition is up or down a half step\textsuperscript{59}; the direction of the transposition will be determined by the new key signature, the second parameter. An example of transposition is given in Figs. B.33.

When transposing, one should always use relative accidentals, activated by the symbol $Ar$ in the header (cf. Section B 6.1 for a detailed explanation of this notation).

When transposing an entire score of same-type instruments, it is of course easy to obtain both the original score and its actual sound, as shown in Fig. B.34.

B 5.5.2 Transposition of Individual staves

In orchestral scores, you usually have individual ‘transposing instruments’, i.e. instruments that are denoted (and, in fact, played) in a pitch different from what they sound: clarinets in E or B flat, Oboes in A (oboe d’amore) or F (oboe da caccia), and similar. For that purpose PMX now provides the option to transpose individual staves.

The instructions given by Don Simons himself on introduction of PMX 2.613 are so clear that it is best to quote them literally here:

```
"The syntax is an extension of the existing full-score transpose/key change command "K". Just follow "K" with "i", then instrument number, then the same parameters as before for transposition amount and new key signature:

   Ki[instrument #][+/-][trans amt.][+/-][new key]
```

\textsuperscript{59}This artifact is needed to differentiate a half-note transposition from a mere key change without transposition (cf. Section B 5.4) .
For more than one instrument, you may immediately repeat everything after “K”. The command must either be at the start of the score (right after setup), or if later, must be preceded by a normal (full score) key change command K+0[+/-][new key]. (Remember that later full-score transpositions are not allowed).

An important function is worth repeating/emphasizing:

- It is usually easiest to put the new K command where the old one was: right before the start of the actual score, i.e. in the ‘header’. But it is now also possible to enter it later, albeit with the (somewhat cumbersome) addition of a ‘dummy’ full-score key change command (as explained above).

This can indeed occur in practice, e.g. when the oboe player switches to English Horn in midstream!

**B 5.6 Titles, and text above and below a system**

A *title block* with up to three elements can be defined in the header (cf. p.22):

- **Ti**: an instrument name. It is set *above* the title, and left-justified,

- **Tc**: the composer’s name. It is set *below* the title and right-justified,

- **Tt**: a title for the whole piece. It is centered on the page.
Each of these commands is to be followed by a text string, *on a separate line*. This text must be fully contained in one line of PMX coding; but the text for the title can be spread out over several lines in print by using the standard \TeX line break symbol `\`.

Extra vertical space can be added between the whole title block and the top system by appending to \texttt{Tt} (without space) a one- or two-digit number representing this space, in units of `\internote`. This works only if \texttt{Tt} is the final title block element entered. Thus, although the 3 parts of a title block can be given in any order, it is good practice to use the sequence $\texttt{Ti} \rightarrow \texttt{Tc} \rightarrow \texttt{Tt}$.

When you want to generate parts from a score with \texttt{scor2prt} (cf. Section C 1), you should omit \texttt{Ti} with an associated instrument name, because each separate instrument usually has a different name. Rather, \texttt{scor2prt} will automatically supply the appropriate instrument names, printing the name of each instrument at the upper left of the first page of each part; the names used here are the ones specified in the preamble of the PMX file for the score. If you do have \texttt{Ti} together with an instrument name in the main score, \texttt{scor2prt} will print that name identically to all the parts — which is usually not what you want!

---

\textsuperscript{60}PMX does this together with the adjustment of other vertical spaces when fixing the general page layout. In some rare cases, the result may not be quite what you expect, in particular when you haven’t used the \texttt{Ae} option to equalize inter-system spacing (cf. section B 6). In such cases you might want to use inline \TeX — take a look at the example in Section E 2.1 for how this can be done.

\textsuperscript{61}\texttt{scor2prt} does this by inserting into the part `.pmx` file a \texttt{Ti} and the appropriate instrument name as taken from the preamble of the main score. Knowing this may be useful if you want to manipulate this process, e.g. to change the vertical shifting of a particular instrument name (cf. Section E 2.1 for details).
The symbols h (for header)\textsuperscript{62} and l (for lower) can be used to introduce text either above or below a system. These symbols must be placed in the first column of an input line (not necessarily the beginning of a block), and followed by a blank or – for h only – a signed integer. In the latter case, this integer is a vertical shift, in units of \texttt{\textbackslash internote}.

The text string, which again must lie on a line of its own immediately following the symbol, will be printed above or below the top staff in the first bar of the block where it is entered. Figs. B.35 and B.1 show examples.

**B 5.7 Page numbering and page headers**

**B 5.7.1 Page numbering**

By default, PMX does not number the pages of a score, even when it has more than one page. You can, however, switch on page numbering by using the symbol P (for Page). Placed at the beginning of an input block anywhere within the PMX body of input, and followed optionally by a page number and/or by l (for left) or r (for right), this will

1. start page numbering at the current page\textsuperscript{63} with the number given as optional argument. Further page numbers are then incremented automatically starting with this number. If P has no number argument, the default is page 1. Thus, P and P1 are equivalent),

2. put the page numbers to the top left or right of the first of the pages to be numbered, after which the number positions will alternate. If no l or r is given, PMX puts the page numbers on the right on odd-numbered pages, on the left of even pages.

You can have as many P symbols in a score as you like; but there is no command to switch off page numbering again once it has been switched on.

**B 5.7.2 page headers**

There is one more option, c (for centered header), available with the P symbol. This option defines a text string to be printed at the top of every page after the first. The notation is as follows:

- The normal usage is Pc"[text string]". The text string enclosed in the quotation marks must be a single line of text. It may contain blanks, but no \textbackslash \). If the string consists of blanks only, that is what will be printed: a blank page header.

- If the text string does not contain any blanks, you may omit the quotation marks, but you must then start the text immediately after the c with no space.

- If there is a blank after Pc, no quotation marks, and no further text, the text printed as the header will be the instrument name entered with the symbol Ti, as described on p.B 5.6.

\textsuperscript{62}This usage of the h symbol is not to be confused with that for the page size (see Section B 6.2)! Although both can occur together in the header, they are in fact, different PMX symbols, distinguished by their differing syntax.

\textsuperscript{63}The ‘current page’ is the page where PMX puts the music currently processed at the point where it encounters the symbol P.
c must be the last option in the P symbol.

The P symbol and all its options will be ignored when making parts from a score using `scor2prt`, since page numbering will usually be different in the score from that in the parts. Page numbering – and centered headers – for parts can, however, be initiated independently with another mechanism. For the details on this, refer to Section C 1.
Oboe I

Concerto in d minor  
RV535

Antonio Vivaldi (1678-1741)

Largo

\[ \text{Figure B.35: Titles, and text above and below a system} \]
B 5.8 Layout: line, page, and movement breaks

The parameters \texttt{npages} and \texttt{nsystems} of the preamble (cf. Section B 3) determine the layout of the score, either by fixing the total number of pages and systems, or by setting the average number of bars per system. With no further instructions, \textsc{Pmx} will attempt to distribute all the music evenly over the total number of systems, and then spread the systems evenly over the specified number of pages, either specified or determined by default (with \texttt{npages=0}).

You can change the average ‘crowdedness’ by changing the preamble parameters. Once this is satisfactory, you can exercise finer, more local control by forcing, i.e. manually inserting, line or page breaks at exactly the places you want them to be. You may also want movement breaks, and \textsc{Pmx} has a convenient way of doing that, in the process dealing easily with such ‘housecleaning’ chores as indenting the first system of the new movement, reprinting the time signature, resetting the bar number counter, and other details to be discussed below.

It is worthwhile, however, to quote Leslie Lamport at this point:

"Don’t worry about line and page breaks until you prepare the absolutely final version"


This is especially true for a \textsc{Pmx}-generated score. Until the final edit, you should always use \texttt{npages=0}. Once you specify any forced line break, you can no longer use \texttt{npages=0}, and you will have to specify values for \texttt{npages} and \texttt{nsystems}. And because — as we’ll soon explain — a movement break is a sub-option of the linebreak command, you cannot set movement breaks either until the final edit.

But unless your score is just a few pages long, you probably will want to set some line, page, or movement breaks in the end. When you are ready, always first pick some reasonable values for \texttt{npages} and \texttt{nsystems}—perhaps the numbers that finally came out when you used \texttt{npages=0} during the main input process. Then one way to proceed is to start at the beginning, forcing breaks where desired. The most common places would be at movement breaks, and at places associated with volte or repeats. Recompile and view the result after each new line break is inserted. You may need to adjust the total number of systems or even pages depending on how crowded is the remainder of the score, after the last forced line break.

The other way to proceed is to begin by inserting any obvious page breaks, especially if you are laying out a booklet with facing pages and want page turns to come in musically convenient places.

Here are the rules for inserting forced line, page, and movement breaks:

- Line breaks can only come at the start of an input block. To force a line break at the start of the \textit{n}\textsuperscript{th} system, enter \texttt{L[n]}. \textit{n} must obviously be greater than 1 and less than or equal to the value of \texttt{nsystems} as specified in the preamble.

- Page breaks can only come where there is already a line break. To force a page break at the start of the \textit{m}\textsuperscript{th} page and the \textit{n}\textsuperscript{th} system, enter \texttt{L[n]Pm}. \textit{m} must obviously be greater than 1 and less than or equal to the total number of pages.
• Movement breaks can only come where there is already a line break. To force a movement break at the start of the \textit{n}th system, enter \texttt{L[n]M}. If in addition you want a page break at that point, use \texttt{L[n]P[m]M}.

The movement break symbol \texttt{M} has several optional parameters. They can be used in any combination and should follow with no spaces.

- \texttt{M+k} adds $k$ \texttt{in\textit{ernote}} units of vertical space between movements.
- \texttt{Mi[,]} resets the first-line indentation of the new movement to \texttt{fracindent=.x}, where \texttt{x} is some decimal number (cf. Sec. \texttt{B 3}).

\begin{quote}
You must write the fracindent number exactly in the form \texttt{x}, i.e. \textit{without} the leading 0. If, e.g., you write 0.2 instead of .2, \texttt{PMX} will complain with an error message, asking for a decimal number!
\end{quote}

- Beginning a new movement break, the bar number is reset to zero by default, whereas \texttt{Mc} continues bar numbering.
- \texttt{M[n]} changes the number of instruments to \texttt{n}. \texttt{n} must not exceed \texttt{ninstr} as set in the preamble. If \texttt{n} exceeds 9, it must be \texttt{preceded} with : (colon). This should be followed (without blanks) by a sequence of \texttt{n} instrument numbers in bottom-up order, again preceded with : if bigger than 9, then a sequence of clef symbols, one for each staff of each instrument starting from the bottom. An instrument’s numeral is simply its position in the original sequence (in the preamble); these numerals can be permuted as desired, but the sequence of clef symbols should, of course, match the new sequence of instruments.
- \texttt{Mr+/Mr-} reprints/suppresses reprinting the instrument names at the beginning of the new movement. The default is to print them only if the number of instruments changes.

It should be emphasized here that the number of instruments in a new movement can never exceed the original number of instruments, although it is permissible to increase it after it has been decreased, as long as it doesn’t exceed the original number; in other words, and to give an example: a sequence quartet — solo — duet, e.g., is possible, while the reverse sequence is not. There are two ways to circumvent this restriction:

1. start the score with a dummy page — to be discarded later — containing the maximum number of instruments. On the second page, start a new movement with the desired number and sequence of instruments for the first movement,

2. make separate \texttt{PMX} files and concatenate them afterwards (cf. Section \texttt{B 1.1} or \texttt{C 4} on how to do this).

Immediately after a movement break, any desired meter changes, key changes, or text can be entered as already described in Sections \texttt{B 5.3}, \texttt{B 5.4} and \texttt{B 5.6} (p.61 ff.).

The effect of movement breaks is illustrated in the example \texttt{G 2.1} in the appendix.
B 5.9 Bar Numbering

By default PMX places a bar number above the first bar of the top staff in every system, and there is no provision in PMX to change this default. But it is easy to change this by using a straight MusiXTeX command:

- if you want no printed bar numbers at all, add a line in the header containing
  \nobarnumbers\
- if you want bar numbers printed over every bar, add a line containing
  \barnumbers\

in the header or at the beginning of an input block.

This is an example of a type 2 inline \TeX command, cf. Section B 8. As is explained there in detail, PMX will place a type 2 inline \TeX command at the beginning of the .tex file produced, regardless of where in the .pmx file the command appears. This means that you cannot change this type of bar numbering in mid-stream.

If you do want to change the bar numbering somewhere in the middle of a score, use a type 3 inline \TeX command: if, from the current point in the score, you want

- no printed bar numbers, add a line with
  \nobarnumbers\
- bar numbers printed over every bar, add
  \barnumbers\
- bar numbers above the first bar of every system (the PMX default), add
  \systemnumbers\

at the beginning of the current input block.

Furthermore, you might want to change the number itself. This occurs most frequently when you have a pickup bar at the beginning of a repeat; some – including some commercial publishers – do not include this pickup bar in the number count, whereas PMX does so by default. To adjust this, use the type 3 inline \TeX command \\advance\barno-1\relax\.

MusiXTeX provides still more elaborate bar numbering schemes, such as a number over every $n$th bar. For details refer to the MusiXTeX 1.15 manual.

B 6 Some general options and technical adjustments

B 6.1 Global options

PMX has many layout parameters with default settings. Some of these can be changed by the user; for that purpose there are several symbols, all beginning with A. Many of these are usually included in the header, but some may be entered at the beginning of any input block. Table B.8 lists these options alphabetically; the following subsections, grouped by types of settings, describe them in detail. When using several of these commands, it is not necessary to enter them all on separate lines; rather, you can concatenate them, in any order, with just a single A, e.g. AdI2.3p+hlbr.
# B 6. SOME GENERAL OPTIONS AND TECHNICAL ADJUSTMENTS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aa[x]</td>
<td>sets space before first note in a bar</td>
</tr>
<tr>
<td>Ab</td>
<td>makes accidentals big</td>
</tr>
<tr>
<td>Ad</td>
<td>puts dots in lower voices below the line</td>
</tr>
<tr>
<td>Ae</td>
<td>equalizes inter-system spacing</td>
</tr>
<tr>
<td>AI[x]</td>
<td>changes default Interstaff spacing for the whole score</td>
</tr>
<tr>
<td>Ai[x]</td>
<td>changes default interstaff spacing for the current page only</td>
</tr>
<tr>
<td>Ak</td>
<td>adjusts rest heights in 2-voice staves depending on context</td>
</tr>
<tr>
<td>An[x]</td>
<td>sets filenames generated by scor2prt (cf. Section C 1)</td>
</tr>
<tr>
<td>Ap[further suboptions]</td>
<td>enables PostScript type K slurs (cf. Section B 4.7)</td>
</tr>
<tr>
<td>Ar</td>
<td>switches accidentals to relative</td>
</tr>
<tr>
<td>AR[filename]</td>
<td>inserts a text file in the PMX source file (cf. Section B 1.1)</td>
</tr>
<tr>
<td>As[ns]</td>
<td>enables PMX Spacing algorithms to small-font staves</td>
</tr>
<tr>
<td>As</td>
<td>makes accidentals small</td>
</tr>
<tr>
<td>At</td>
<td>activates special xtuplet brackets (cf. Section B 4.6)</td>
</tr>
<tr>
<td>Av</td>
<td>spreads systems vertically over all of an unfilled page when ‘on’</td>
</tr>
</tbody>
</table>

Table B.8: Symbols beginning with an A (global options)

## B 6.1.1 Accidentals

By default, big accidentals are used unless regular spacing doesn’t provide enough room. Thus the default behavior may cause a mixture of big and small accidentals, and in fact is not recommended. Rather, the user is advised to make his choice:

- **Ab** makes all accidentals big; this is usually preferred,
- **As** makes all accidentals small.

As was mentioned earlier (cf. Section B 5.5), PMX by default uses the so-called ‘absolute’ notation for inputting accidentals, i.e. the notation explained in Table B.5. In this notation, there is an obvious one-to-one correspondence between accented characters in the PMX source file (s, f, n, ss, ff) and the printed characters. This has a disadvantage, but it only surfaces if you write the PMX file in one key and later transpose it using the K option described in Section B 5.5. For example, suppose the original key is F, with 1 ♭ in the signature, and you enter bn. Now if you transpose to the key of C with K-3+0, that note will be printed explicitly as F natural, while it should be F sharp.

The remedy for such problems is to use another notation, aptly named ‘relative’ accidental notation, used by some musicians and some publishers in some countries. In this notation, a sharp, flat, or natural sign denotes an alteration by a half-tone up, down, or none **relative to the note which would otherwise be indicated based on the signature of the piece**. For example, in D major, with 2 ♯s in the signature, an F natural would be denoted not as fn but rather as ff, while in d minor, with one ♭ in the signature, a B natural would be input not as bn but...
rather as bs. Similarly, ♮ accidentals refer to the notes prescribed by the signature; e.g., in G Major (1 ♯ in the key) a gn sounds as a g, while fn sounds as an F sharp.

For automatic transpositions to function properly, it is necessary that the PMX source for the score to be transposed use the relative accidental notation. This is effected by Ar:

- Ar directs PMX to interpret accidentals in relative notation.

B 6.1.2 Layout details

When the note head of a dotted note is on the line (e.g. for a g in a violin clef), it is customary to place the dot slightly above the line. In staves with two voices, there is an alternate custom, viz., to place the dot above the line in the upper voice and below the line in the lower voice. PMX’s default is the first possibility; dots for both voices above the line, but it can be overridden:

- Ad causes the dot – for notes that are on a line – in the lower voice of any two-voice staff to be placed below the line.

By default PMX inserts a horizontal gap of 1\elemskip between a bar line and the first note in the bar. This can be changed globally:

- Aa[x] sets the space before the first note in every bar to to x units of \elemskip.

In PMX it’s not yet possible to specify a smaller font for selected staves. But it can be done using an inline TeX command (For details, cf. Section B 8 and Section E 2.4). If you do this, then you ought to use the AS option:

- AS[ns] . The additional sequence of ns is mandatory here. There must be exactly as many of the characters 0 (zero), s, - (minus), t as there are staves in the score (nstaves, cf. Section 3). These characters designate the font size of the staff:
  * 0 if the font size of this staff is normal,
  * - or s if it is small,
  * t if it is tiny.

This tells PMX to modify some horizontal spacing decisions to account for the smaller font size.

B 6.1.3 Vertical spacing

a) Spacing of staves within a system:

If there is more than one staff in a system, PMX computes the vertical spacing between the staves automatically. However, the algorithm isn’t fully robust, and the result may not be pleasing. To change the spacing between the staves within a system, you can apply a scale factor x to the default, either for the entire score, or for the current page only:

---

64 Note that this makes logical sense only if used in the header, and accordingly there is no way to ‘undo’ this decision further down in a score.

65 The sequence of staves is from bottom to top, as described in the preamble. Cf. p.22.

The possibility of using s and t in the sequence of ns is a new enhancement feature of PMX 2.613.
• $\text{AI}[x]$ multiplies the default interstaff spacing, $\text{\interstaff}$, by the decimal number $x$ for the entire score. This option should be placed in the header.

• $\text{AI}[x]$ multiplies the default interstaff spacing by the decimal number $x$ for the current page only. This option can be placed at the beginning of any input block (including the first), and overrides $\text{AI}[x]$.

Note that if there is more than one system on a page, decreasing the interstaff spacing will increase the space between systems, and vice-versa.

b) Spacing of systems on a page:
MusiXTeX normally draws a virtual box around each system and inserts equal vertical space between these boxes. When objects protrude above the top staff in a system – such as the note $c46$ in the violin clef – or below the bottom one, this can lead to unequal spacing between the bottom staff line of one system and the top staff line of the next. You may prefer that this vertical spacing be constant for the whole page:

• $\text{Ae}$ ensures that the spacing between the bottom staff line of one system and the top staff line of the next is constant for any one page.$^{66}$

c) Sparsely filled pages:
Sometimes it may happen that a page contains just a few staves altogether, and then the white space between staves on such a page becomes excessive. In such cases – specifically, if $\text{\interstaff} > 20\text{\internote}$ – PMX will, instead of the equal spacing described above, group all systems near the top of the page. This can, however, be changed by the user:

• $\text{Av}$ will suppress the grouping near the top, and ensure that systems will always be spread vertically regardless of how much white space is left between systems.

Note that $\text{Av}$ acts as a toggle; the second time it is issued, the behavior will revert to the default.

d) Vertical position of rests in 2-voice staves:

• $\text{AK}$ activates special rules for vertical positioning of rests in two-voice staves.

This general option is rather subtle; its explanation by Don Simons follows:

“Without this option, rests in two-voice staves have default positions based on a simple rule that is not context-sensitive: those in the lower voice (the one before $//$) are $4\text{\internote}$ heights below their single-voice default positions, and those in the upper voice are $2\text{\internote}$ heights above the single-voice default.

“The $\text{AK}$ option invokes a set of context-sensitive rules to set the default position. The baseline rule is to align the rest in a horizontal line with the

$^{66}$It is generally advised to use the $\text{Ae}$ option by default, unless one has a specific reason not to. When using this option, you may Nevertheless, in certain cases you may want to force more vertical space between certain systems. There is a $\text{TEX}$ macro, called $\text{\spread}[x]$, that can be inserted anywhere in the system before the desired wider gap. Its argument $x$ is the desired extra space, in units of $\text{\internote}$ (cf. Section B 8).
next following note in the same bar. If there is no following note in the bar, then the rest is aligned with the last note before the rest.

“If there are simultaneous rests in both voices, the old rule is applied.

“The AK option is a toggle: it may be used at the start of any input block to turn these special rules on (if previously off) or off (if previously on).

“When the AK option is on, it only affects places where there are two voices in a staff.

“When any user-defined tweaks on the height of a rest will supersede the option for that particular rest, i.e. the tweak will be applied relative to the single-voice default position.”

Sometimes you may want the rest to be aligned with the previous note (the one to the left of the rest), rather than the following note (the one to the right of the rest). This can be achieved simply by appending an L (for ‘Left’) to the rest symbol, i.e. by writing ‘rL’.

B 6.1.4 PostScript type K slurs, ties and hairpins

The use of PostScript slurs, ties and hairpins circumvents most of the shortcomings of the font-based counterparts. Of the two available packages, only the type K slurs are explicitly supported by PMX. There is an option, Ap, to enable the use of this package.

- Ap activate Type K PostScript slurs, ties and hairpins.

Ap may also be entered at the beginning of any input block to activate certain suboptions controlling shapes and positions of the slurs and ties that are to be used. The first few of these options affect vertical positioning. When Type K PostScript slurs are activated with Ap, slurs and ties will by default not have their vertical positions tweaked to avoid tangencies with staff lines.

To control this type of adjustment, use one of the following suboptions to Ap:

1. -s activates automatic slur height adjustment,
   -t activates automatic tie height adjustment,
   -s/-t deactivate the corresponding height adjustment.

2. A second suboption, +c/-c, will increase or decrease the default curvature of the Type K slurs. The result of such commands is cumulative, and more than one suboption +c or -c may be used in a single command. If the cumulative change goes above HH or below f, a warning will be issued, the default will be set to HH or f, and processing will continue.

Local curvature options in individual slur commands (cf. Sec. B 6.1.4) will take precedence over the global default, but will not change the default.

A special local option, n (‘normal’), is of the ‘undo’ type: it will cause that particular slur, no matter what the global default happens to be, to have the normal curvature, i.e. between f and h.
3. A third suboption, \( l \), changes the appearance of slurs and ties that extend across line breaks:

- Every slur/tie at a line break is automagically broken into two separate ones; no additional PMX slur start or ending commands are required. Vertical and horizontal tweaks for the end of the first segment and Start of the second segment are entered as options in the normal command that starts the slur/tie:
  
  (a) the option for the end of segment 1 starts with \( s \) (for severed or split), then the usual one or two signed numbers, then a second \( s \) and one or two more signed numbers for the start of segment 2,
  
  (b) the usual curvature options \( h, H, HH, f \), if included in the starting command for a line-break slur, will apply to segment 1, and to segment 2 if in the closing command. If the slur/tie does not come at a line break, the special position tweaks (those after the \( s \) option) will all be ignored, and the curvature tweaks on the closing note take precedence.

The \( l \) option can also be invoked globally (i.e. for the entire score) by specifying \( Ap1 \), instead of only \( Ap \), in the header (recommended).

The third suboption concerns only line-break ties (it does not affect slurs):

- \( +h/-h \) activates/deactivates the use of special half-ties for the second segment of line-break ties after the line break. These are horizontal at their left end, and are only used if the required segment is shorter than 15 pt.

Note that this option is somewhat incompatible with the \( l \) option, and they should not be used simultaneously.

### B 6.2 Page Size

The default page size\(^67\) is 740 pt \( \times \) 524 pt (10.3” \( \times \) 7.3”, or 261 mm \( \times \) 185 mm). To change the height or width, use the special symbols \( h[n][u] \) or \( w[n][u] \) in the header. Here \( n \) is a decimal number for the new size, and \( u \) defines the units: \( i \) for inches, \( m \) for millimeters, and \( p \) for points. if no unit name \( u \) is given, PMX uses points as the default.

This command can be used together with \( \%\% \) or \( \%! \) (see Section C 1) to give the parts made by \texttt{scor2prt} different page sizes than the parent score.

\[\text{This usage of the } h \text{ symbol is not to be confused with that for a line of text above a system (see Section B 5.6)! Although both can occur together in the header, they are in fact, different PMX symbols, distinguished by their differing syntax.}\]

\(^{67}\text{This means the extent of the print on the page. The empty empty top, bottom, left and right margins depend on the paper/printer used; they cannot be altered within PMX.}\]
B 6.3 Stem direction of bass notes

By default PMX makes stems go up for middle-line D’s in bass clef, but down for notes on the middle line of all other clefs. If you want middle-line bass-clef notes also to have downward stems by default, enter the symbol B in the header.

B 6.4 Horizontal Spacing

Minimum Spacing between Notes

PMX does some special, complex analysis to adjust horizontal spacing in crowded systems. By default, the minimum space between consecutive noteheads is 0.3 notehead widths. If you want to change this value to some other fraction, enter \( \frac{w}{n} \), where \( n \) is a single digit between 1 and 9, giving the minimum spacing, in tenths of a notehead width. This adjustment is very rarely used.

Extra Horizontal Shifts and hardspace

PMX will usually provide satisfactory horizontal spacing. However, there may be some occasions where you will want to adjust it manually. A symbol starting with X controls one of two available types of horizontal adjustment:

1. a shift moves one or more characters in the current voice but does not affect any other spacing anywhere,
2. a hardspace, by contrast, is a fixed amount of space inserted at a particular time. The horizontal positions of everything in all staves in the system will be adjusted to accommodate the added space while maintaining vertical alignment.

Here are the syntactic possibilities:

- \( \text{XS}[x] \) (for Shift). This command shifts the the next character; \( x \) is a decimal number, giving the amount of shift, in units of a notehead width.
- \( \text{X:}[x] \) or \( \text{X}[x] \): initiates a “group shift”, which operates like \( \text{XS}[x] \), except that everything from the insertion point onward in the current voice is shifted until a termination symbol (“shift end”) is encountered.
- \( \text{X} \): ends a “group shift”.
- \( \text{X}[x] \) (without any S or :) is a so-called hard space. It inserts the specified space, given in units of notehead widths, at the present point in all staves of the system. If \( x \) is negative, space will be removed.

Because horizontal spacing in parts will usually differ from that in the score, scor2prt (cf. Section C 1) will by default copy only the shift commands \( \text{XS}[x] \), \( \text{X:}[x] \) and \( \text{X} \) into the parts, but not the hardspace command \( \text{X}[x] \).

This behavior can be overridden using the methods described in section C 1. There is, however, an alternative method that helps to keep PMX score files neat and readable: add the suboption B or P (without space, as usual) to the X symbol:
• B (for Both): with this option, the spacing symbol is to be applied to both score and part,
• P (for Part): with this option, the spacing symbol is to be applied to the parts only.

B 7 Macros

A PMX macro is a single symbol that stands literally for any any string of characters that may occur in the input file.\(^{68}\) Macros may be useful to save tedious typing and disk space, and make PMX source text more readable if you need to repeat the same string many times in a score.

You may define up to 20 macros in any single PMX input file. There is no practical limit to the length of the character string that a macro represents, but of course it must respect the PMX rules on the length of input lines and what must go on separate lines.

Macros only exist within the PMX input file where they are defined; you can define a completely new set of macros in another PMX input file. Conversely, if you want to re-use the macros of file myopus1.pmx in myopus2.pmx, you need to copy them explicitly from one file to the other.

There are two distinct ways to define and record a new macro or redefine an existing one:

1. to Record a Macro: type MR\([n]\), followed by a space, as usual. \(n\), the “name” of the macro, is an integer between 1 and 20. Everything you type after this will be processed normally, at that point in the input file, as well as stored, until you end the macro by entering the symbol M.

2. to Save a Macro: type MS\([n]\). Everything you type after this will be recorded (saved) as you enter it, without processing the PMX code. When saving macros, it is usually best to put them in the header, for readability’s sake.

Whenever you would otherwise need to re-enter the same character string, you can simply Play back the macro by typing MP\([n]\).

Macros can be redefined at will; PMX will issue a warning whenever this occurs.

When you use macros and want to make separate parts (cf. Section C 1), some care is necessary: scor2prt will transfer a MR macro only into the part where it originated, but will transfer MS macros into all parts.

So if you want to make macro 1, which you will use in several or all parts and then use it immediately in the part in which you write it, dont use simply MR1, but rather

\[
\begin{align*}
\text{MS1} \\
\text{MP1}
\end{align*}
\]

\(^{68}\)Note to seasoned programmers: PMX macros are not really macros in the usual sense, but merely string abbreviations; they do not allow for variables.
Fig. B.36 contains an example of the use of macros.

% macro 2 saved:
MS2 e4 f g2 o_ | M
% Bar 1 (macro 1 recorded and processed):
Rl MR1 c45 d e c | M
% Bar 2 (macro 1 played):
MP1
% Bar 3 (macro 2 played):
MP2
% macro 4 saved:
MS4 c4 g c2 o_ M
% Bar 4 (macro 2 played):
MP2
% Bar 5 (macro 3 recorded and processed):
MR3 g85 a g e4 c | M
% Bar 6 (macro 3 played):
MP3
% Bar 7 (macro 4 played):
MP4 |
% Bar 8 (macro 4 played):
MP4 Rr /
%

Figure B.36: “Frère Jacques” (usage of macros)

B 8 Inline \TeX{} commands

It may be worthwhile to reiterate at this point the very different intentions of the software ingredients which make up the \textbf{PMX} tool. It is, in fact, a hierarchy:

1. At the base of it all is \TeX{}. \TeX{} (written by Donald E. Knuth) is a general and extremely powerful tool for typesetting high-quality documents on a computer that are ready for printing. But \TeX{} is much more than that: it is a full-grown and powerful computer programming language with which – if you are a \TeX{}pert – you can do almost anything.

There is, however, a price to pay for this power: while you don’t have to be a professional computer programmer to use \TeX{}, a pretty good familiarity with the basics of

\footnote{“Its emphasis is on art and technology, as in the underlying Greek word.” (quote from the introduction of \textit{the \TeX{}book}, the authoritative book on \TeX{}, by Donald E. Knuth.}
programming is unavoidable.

2. MusiXTeX (written essentially by Daniel Taupin) is a ‘macro’ for \TeX\footnote{To be precise, MusiXTeX consists of a very extensive set of \TeX\ ‘macros’ and music typesetting fonts.}. That alleviates most of the \TeX\ programming tasks for the special purpose of typesetting music. Thus, while it helps to have a programming background, such a background is not absolutely essentially for someone who just wants to write a few simple music scores.

3. PMX, the ‘preprocessor’ for MusiXTeX written by Don Simons and described in detail in this tutorial, is yet another level of abstraction on the long way from the musical ideas that make up a score to getting ink on the paper at certain points. The whole purpose of PMX is to make typesetting of music with MusiXTeX/\TeX\ accessible to people with essentially no prowess in computer programming.

As a consequence of this hierarchy, PMX is vastly simpler to use than MusiXTeX, which in turn is simpler than straight \TeX. But the converse of this is also true: there are elements in MusiXTeX that are not directly accessible with the PMX language, just as you cannot unleash the full power of \TeX with MusiXTeX alone. So to allow the user full access to ‘lower-level’ MusiXTeX or \TeX – while maintaining its higher-level notational simplicity – PMX supports the inclusion of \TeX\ commands either directly in the PMX input file or via an external file.

### B 8.1 Including TeX Commands in the .pmx source file

The normal way for a PMX user to insert user-defined \TeX\ code is to enter it directly in the .pmx input file; this is what is referred to as “inline \TeX” in this tutorial. There are four ways to do that; these differ mainly in where the \TeX\ code will appear in the PMX output .\TeX\ file. The first 3 of these have \,, \\\textbackslash or \\\textbackslash (1, 2 or 3 backslashes) as a starting symbol, then comes a sequence of \TeX\ commands, then another \ (backslash) as a terminating symbol, followed by the usual space. All three of these must be contained fully in one line of input, limited by default to 128 characters (cf. Section D 1); but this one line can contain any number of concatenated \TeX\ commands.

The above description needs clarification on a subtle point: the starting symbols \,, \\\textbackslash or \\\textbackslash are not really PMX symbols in the sense used throughout this tutorial, in that there would be ‘white space’ after them, but the \TeX\ command(s) follow them without a blank. This has to do with the fact that all \TeX\ commands – called ‘control sequences’ – start out with a \ . And thus the last of the starting backslashes (the only one for type 1, the second one for type 2, the third one for type 3) counts as the \TeX\ control character of the following \TeX\ command. For example, if you wanted to define a new \TeX\ command \texttt{Myspace} as a type 1 inline command, you would write

\begin{verbatim}
\def\Myspace{hskip{10mm}}\end{verbatim}

whereas if it is to be a type 3 inline command, it would read

\begin{verbatim}
\def\Myspace\relax{hskip{10mm}}\end{verbatim}
in other words: the starting symbol itself also provides the first \ of the \TeX command (control sequence).

This convention can also be seen clearly in the examples given below.

Here are the characteristics of the first three types:

Type 1 ( \ ... \ ) : a Type 1 string usually serves to define or execute a command that is only needed locally for the current line of music, right after the place where it is entered in the .pmx file. \PMX inserts it in the resulting .tex file right before the \MusixTEX command for the next note or rest. Multiple type 1 strings associated with the same note or rest are allowed, although the total length may not exceed 128 characters. So there is generally no reason not to combine all \TeX commands for a single note into a single type 1 string.

If an inline \TeX string contains a new \TeX definition, this definition is in general not local; depending on circumstances, it can remain valid throughout the score and for all instruments unless redefined somewhere! This is true for all types of inline \TeX, Type 3 in particular. So be careful!

Type 2 ( \ \ ... \ ) : a Type 2 string usually serves to define or execute a command that is needed throughout the entire score. It will appear near the top of the .tex file, right before the \startmuflex, regardless of where it appears in the .pmx file.

Type 3 ( \ \ \ ... \ ) : Type 3 strings will appear in the .tex file right before the \xbar or \alaligne of the current input block, i.e. before its first bar line. Thus a Type 3 string is typically used when you want to redefine a previously defined \TeX command, to be valid from a certain point on, but not before.

Type 4 : The fourth way of inserting inline \TeX in a .pmx file is different from the first three in several ways:

- it is started by the character sequence --- (triple dashes), on a line by itself, as the very first line of the .pmx file,
- it is terminated by the next appearance of the same character sequence, --- (triple dashes), on a separate line,
- it can contain any number of lines with \TeX code, although every single of these lines is again restricted to a total of 128 characters,
- the lines of \TeX code between the opening and terminating triple dashes will be copied literally to the very top of the .tex file.

So Type 4 can also be (ab)used to copy comment lines literally from the .pmx file to the .tex file. This is convenient for documentation purposes.
Another difference of practical importance is the way in which `scor2prt` handles these types of inline TeX when making separate parts (cf. Section C.1): types 2-4 will be copied into all parts, while type 1 only goes into the part in which it appears in the original .pmx file.

To understand better the differences between these 4 types on inline TeX, it may be useful to study the basic structure of the TeX file `inlinesample.tex`, given in Table B.9, produced from the PMX input file `inlinesample.pmx`, given in Fig. B.37. In doing so, you may want to look at the box on p. 83.

\begin{verbatim}
% beginning of inlinesample.pmx
---
% This sample demonstrates the use of all 4 types of inline TeX
---
% PREAMBLE:
% 
1 1
1 2 0 0
0 0
1 2 16 .0
%

t
./
% BODY:
% HEADER:
\w80m
\nobarnumbers\%
\begin{musicinput}
\zcharnote{-2}\{(PED}\ a D"\DEP"+3+15 | / \%
\end{musicinput}
\barnumbers\ c Rb /
% end of inlinesample.pmx
\end{verbatim}

Figure B.37: A sample for the use of inline TeX
This sample demonstrates the use of all 4 types of inline TeX
\input musixtex
\input pmx
\input musixmad
\smallmusicsize%
\nopagenumbers
\tracingstats=2\relax
\hsiz=227pt
\vsize=740pt
\def\nbinstruments{1}
\setstuffs11
\setclef10
\setname{
\generalsignature{ 0}%
\parindent 0pt
\elemskip\afterruleskip1.000pt\beforeruleskipOpt\relax
\stafftopOpt\staffbotOpt\Interligne\interstaff{10}\relax
\nostartrule
\readmod{inlinesample}
\nobarnumbers%
\startmuflex\startpiece\addspace\afterruleskip%
\BODY:
% begin of music input:
\znotes\zcharnote{16}\{\titles{2.0}\{}{0}\{}{0}\{}{0}\}\en%
% Bar count 1
\pnotes{4.00}\hu g\en%
\systemnumbers%
% Bar count 2
\alaligne
\pnotes{4.00}\zcharnote{-2}\{\PED}\txtdyn{-2}\{15.0}\{\normtype\it \DEP}\%\hu{'a}\en%
% Bar count 3
\barnumbers%
% Bar count 4
\xbar
\pnotes{4.00}\hl{'c}\en%
\endpiece
\vfill\eject\endmuflex
\bye

Table B.9: inlinesample.tex, as produced from inlinesample.pmx
Here is a short survey of what happens in the file *inlinesample.tex* (cf. Table B.9), produced from *inlinesample.pmx* (cf. Fig. B.37):

- first, after the file name comment, which is inserted automatically by *pmxab*, the Type 4 string (the comment line on the top of the .pmx file) is inserted [line 6],
- then the files *musixtex.tex*, *pmx.tex* and *musicmad.tex* are always read in [lines [7,8,9]. This means that the .tex that PMX produces has access to all the *tex* macros defined in these files, and therefore you can use all of these in your inline *tex* commands,
- next, up to and including \nostartrule [line 23], there are a lot of settings, the details of which need not concern us here,

> The following *tex* macro [line 24], \readmodsamp, tells *tex* to read in the file *inlinesample.mod* (if it exists). The file *inlinesample.mod* may contain any valid *tex* code of your choosing (cf. Section B 8.3 for details on this),
- next [line 25] is the Type 2 string “\nobarnumbers”. Type 2 strings are always inserted at this point of the .tex file,
- \startmuflex [line 26] – never mind what exactly it does – is the actual beginning of the musical code,
- several lines down [line 31] there is an item \hug: this is the half-note g of the first bar,

> line 32 contains the Type 3 string “\systemnumbers”. This changes the bar numbering scheme to the PMX default (cf. Section B 5.9). PMX inserted this Type 3 string right before the next \alaline [line 34],
- line 35 contains the Type 1 string “\zcharnote-2\PED”, which prints the MusiXTex pedal \^o,
- line 37 contains another Type 3 string “\barnumbers”, before the code for bar 3 , inserted right before \xbar in line 39 (coding the bar line). This changes the bar numbering scheme once again.

^oThe ‘end of pedal’ symbol \DEP is coded here in an alternative (perhaps simpler) way by using a PMX dynamics textual symbol (cf. Section B 4.12).
**B 8.2 Denoting pitch in inline \TeX**

When using inline \TeX, you should be aware of a subtle aspect, which can be a hazard:

In MusiX\TeX, you can always refer to a pitch with a number, rather than a letter, and that number denotes exactly the position on whatever staff you are using, counting from the bottom line as 0 .

When you use this for a pitch, it will not be transposable, nor will it respect (or indeed affect!) the memory \PMX keeps on the pitch: remember that \PMX doesn’t interpret any inline \TeX .

You should keep this in mind when designating a pitch within an inline \TeX command.

We close this section with two practical examples and a final remark:

- As was stated in Section B 4.11.3, when using Type M slurs you need to insert the line

```latex
\input musixpss\relax
```

into the header of the .pmx file. This tells \PMX to read in the file musixpss.tex, which contains the macros for Type M slurs.

This is the case of a type 2 inline \TeX string, and thus the input command is added to the .tex file right before `\startmuflex` (which is where it belongs).

- Sometimes one needs clefs other than those defined in \PMX (cf. Section B 3.1). Some special clef symbols (among them no clef at all) are defined in MusiX\TeX and can be invoked by the command `\setclefsymbol` (cf. the MusiX\TeX 1.15 manual, Sec. 2.21.2). The most common of these are octave treble and octave bass clefs, in which a small 8 is attached to either the top or the bottom of the clef symbol. To get a higher octave treble key for instrument number 2, for example\textsuperscript{72}, just add the following type 2 inline MusiX\TeX command

```latex
\setclefsymbol{2} \trebleoct
```

The other 3 octave clef symbols are `\treblelowoct, \bassoct, \basslowoct`, for lower octave treble, higher octave bass and lower octave bass clefs, resp.

More examples can be seen in Section E and in some of the scores of the appendix.

**B 8.3 Putting \TeX Commands in an external file**

\PMX provides one further option for entering an unlimited set of \TeX commands, read into the .tex file by `\readmod` just before the `\startmuflex` line, command, and before any Type 2 inline \TeX strings. Simply put the commands into a text file named `[filename].mod` in the current directory\textsuperscript{73}.

\textsuperscript{72}Matters are more complicated for instruments with more than 1 staff, such as piano or organ, because the number in the command refers to the number of the instrument, not the staff. Cf. the MusiX\TeX 1.15 manual, Sec. 2.27.10 for details.

\textsuperscript{73}This feature is retained mainly for backward compatibility with previous versions of \PMX; it has now been essentially replaced by the 4 options for inline \TeX strings described above.
Chapter C

Special Features

C 1 Making Parts from a Score

Separate parts can be made from a score with the tool `scor2prt` (included in the standard PMX distribution).

As can be seen in the example, the parts files will be named `[yourscorename][n].pmx`, where `[n]` is the number of the instrument, by default; but the names of the files created by `scor2prt` can be changed from within the PMX file for the whole score, the “parent file”. For example, to cause the file for instrument 3 to be named `mviolins.pmx`, include the command `AN3"mviolins"` in the header of the parent file (cf. Table B.8).

In the remainder of this section we describe how to control the layout of the parts separately from that of the score by using commands that are placed in the parent file. This eliminates the need for editing the `.pmx` files for the parts separately. You can accomplish all the editing in the parent file, and then re-run `scor2prt` as required. Thus both the score and the parts can be corrected together, and the parts need not be re-edited each time they are re-generated from the score.

C 1.1 Usage

Since comment lines are by definition disregarded by TeX and by the PMX program itself (`pmxab`, to be exact), they can, with a slight extension of their syntax, be used to transfer information meant for the parts only\(^1\). The following rules are used for this by the `scor2prt` program:

- If a line has `%` in columns 1 and a space in column 2, it is considered a regular comment and transferred as such to all parts.

- If a line has `%` in columns 1-2, both it and the following line will be ignored when making parts. If the ignored line (the second line) has h, l, or T in its first column, then one additional line will be ignored (cf. B 5.6 for the rationale behind this rule).

\(^1\)This is quite a useful method, employed in a similar way by other programs, such as the PostScript language.
• If a line has %! in columns 1-2, these first 2 characters will be stripped, and the rest of the line will be put in the .pmx files for all the parts.

• If a line has %[\textit{h}] in columns 1-2, where \textit{h} is the hexadecimal digit representing the instrument number (1, 2, \ldots, 9, a, b, c), the first two characters will be stripped and the rest of the line transferred to the part for instrument \textit{h} only.

A good example for this usage is the page break(s) in a longer score, where the appropriate page numbers for the parts will be different from those in the full score. For example, to force a line break to system 15 and a page break to page 2 in part 11 only, enter %bL15P2.

The use of the hexadecimal digits a-c in this rule creates a potential incompatibility with previous versions of PMX. To minimize this, the character after “%” will only be interpreted as a part number if it represents a number less than or equal to \textit{noinst}, as given in the parent file; otherwise the entire line will be treated as an ordinary comment, and transferred to all parts.

• In addition to all entries in the header except I options (cf. Sec. C 2, p. 88), the following PMX symbols with all their options will automatically be copied to all parts (unless the previous line starts with %%, of course):

\begin{itemize}
  \item \texttt{R} \quad \text{(bar symbol)}
  \item \texttt{V} \quad \text{(volta)}
  \item \texttt{K} \quad \text{(key change or transposition)}
  \item \texttt{A} \quad \text{(global options)}
\end{itemize}

This is true for \texttt{R,V,K}, in spite of the fact that they can be entered in the first voice of the score only!

C 1.2 The \texttt{S} symbol

• \texttt{S}[n]:

By default the total number of systems in each part will be the same as in the score. If you want to override this, you can do that with \texttt{S}[n], where \textit{n} is the desired number of systems. \texttt{S}[n], if used, must appear in the header of the parent file. Legally, you could use \texttt{S}[n] to reset the number of systems for the parent file itself, which you had just defined in the preamble. That, of course, does not make much sense, but in the form

\begin{itemize}
  \item %!S[n]
  \item %[\textit{k}]S[n]
\end{itemize}

it does, because then it does not affect the full score, \textit{but all parts}, or part \textit{k} only, resp. \texttt{barsant.pmx}, one of the examples given in Section G 1, includes a demonstration of this.
• \texttt{S[n]P[m]}:

\texttt{scor2prt} will also compute how many pages it thinks each part should have, and enter that into the preamble for that part. If you wish to override that, then insert into the parent file \texttt{%3S14P2} for example, which would force the third part to have 14 systems and 2 pages. You cannot override the number of pages without first overriding the number of systems.

• \texttt{S[n]m[k]}:

A musicsize of 20 is the default in all parts. This may be overridden with the option \texttt{m} in the symbol \texttt{S}: e.g., \texttt{%2S15m16}. Again, you must specify the number of systems before setting the musicsize.

<table>
<thead>
<tr>
<th>Keep in mind the distinction among the various usages of \texttt{P}:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• as an option with \texttt{S}, it sets the total number of pages in a part,</td>
</tr>
<tr>
<td>• as an option with \texttt{L}, it forces a page break,</td>
</tr>
<tr>
<td>• as a \texttt{PMX} command on its own, it controls page numbering and centered headings.</td>
</tr>
</tbody>
</table>

C 1.3 Other usage rules

• Inline \TeX strings of Type 2-4 will be copied to all parts, while a type 1 inline \TeX string will only go into the part in which it occurs in the parent file.

• User-defined hardspaces (\texttt{X} without \texttt{:) are by default not copied to parts. There are two ways to change this default:

1. use the options of the \texttt{X} command (cf. Section B 6.4): \texttt{B} causes the hardspace to be used in both score and parts; \texttt{P} puts it into the parts, but not the score,
2. to insert \texttt{x} notehead widths of hardspace into part \texttt{n}, place the symbol \texttt{%[n]X[x]}, on a line of its own, in the parent file.

• As was already noted (cf. Section B 5.7), a \texttt{P} symbol for page numbering in the parent file is ignored when making parts. To initiate page numbering in the parts, use, for example, \texttt{%!P}, with appropriate options added, anywhere within the \texttt{PMX} code representing the first page of the parts\footnote{From \TeX's standpoint the command must occur between the beginning and end of the page on which the numbering is to begin.}. It will often be useful in this case to use the option \texttt{c}, which by default causes the instrument name to be centered in small type at the top of every page after the first.
CHAPTER C. SPECIAL FEATURES

• MIDI commands, i.e., those starting with \texttt{I}, will never be copied into parts, unless they are in a special comment line as just described.

• One function of \texttt{scor2prt} is to condense consecutive bars of rest into a single group of special printed characters with a number above it. Thus \texttt{scor2prt} will automatically insert \texttt{rm} symbols (cf. Sec. B.4.5) into the \texttt{.pmx} files for the parts where appropriate. However, for this feature to work, the \textit{first} full-bar rest in the sequence must have its duration explicitly defined in the parent file, either with a digit or with \texttt{p}. In other words, the feature will not work if the first rest in the sequence inherits its duration from the previous note.

In standard \TeX the rule for comments is: a \texttt{%} character \textit{and all text that follows it on the same line} is treated as a comment, i.e. it is totally ignored by the program. As a consequence of this rule, many users of \TeX{} have a habit of adding comments at the end of short lines of text, instead of writing a whole comment line, starting with a \texttt{%} in column 1. This is fine as far as \texttt{PMX} is concerned; but \texttt{scor2prt} redefines this rule a bit and in general \texttt{will misinterpret comments added in the same line after some legal PMX code.}

\textit{So when you intend to make parts using \texttt{scor2prt}, it is good practice to restrict true comments to lines with \texttt{% in column 1, followed by at least one blank.}}

C 2 Making MIDI Files

Some users of \texttt{PMX} and MusiX\TeX{} may never have heard of MIDI and may therefore be a bit bewildered by this section. So here is a quick introduction:

MIDI stands for “\textit{Musical Instruments Digital Interface}”. It is essentially a standard, consisting of a language and its hardware implementation, designed to produce digitally coded music with a device like a synthesizer or a multi-media computer. If your computer is equipped with the appropriate hardware and software (which is standard on PC’s these days), you can listen to the music encoded in a MIDI file on your computer just as you can read the text encoded in a text file\textsuperscript{3}.

This section describes how to produce such MIDI files when writing a piece of music with \texttt{PMX}. When using this facility, you should be aware both of the general restrictions of MIDI and the special ones that \texttt{PMX} imposes:

\textsuperscript{3}If you are interested in details on MIDI in general, a good place to start is the home page of the “MIDI Manufacturer’s Association”: \url{http://www.midi.org/about-midi/aboutmidi3.shtml}. 
First of all, you should not expect the MIDI files that PMX produces to be anything that resembles a performance of the piece by real, human musicians. While MIDI does make different shades of sound for the different instruments, the lack of any kind of expressivity (rhythmic accentuation, dynamics, vibrato, intonation etc.) does not really do justice to the musical character of acoustic instruments.

So the MIDI file can be hardly more than an acoustic check on whether the notes produced are indeed what you had intended. But for that purpose it can be of great help in coding a score with PMX, so it is generally advisable to produce the MIDI files.

- The MIDI module of PMX does not recognize graces, ornaments, repeats, volte, or segnos. The only ties that are recognized are those using $s$, $t$ or ( ; alone, with no explicit ID number.

- the MIDI file generator does not support changing the number of instruments in mid-stream. You will not see an error message, but the results will be unpredictable.

The MIDI file is produced by PMX concurrently with the MusiXTeX output. To start this, enter the symbol $I$, together with any options (as described below), usually in the header of the input file; but $I$ symbols can appear later in the file as well, but only at the start of an input block.

If the name of the PMX source file is, say, $\text{Mymusic.pmx}$, the MIDI symbol $I$ will cause a file $\text{Mymusic.mid}$ to be written in the directory given in the preamble. If $I$ is used without any options, PMX will use default values for several of the parameters; these default values are indicated below.

Usually, however, you will want to specify some options. They follow the MIDI symbol $I$ immediately, without a space. Sometimes the order of the options matters, so it is generally advisable to adhere to the order in which they are given here:

1. $t[x]$ sets the tempo to $x$ quarter notes per minute. Default is 96. You can change the tempo as often as you like, but only at the start of an input block (as with all MIDI commands).

2. $i[i_1i_2...in]$ assigns MIDI instrument names $i_1,i_2,...,in$ to the staves of the respective PMX instruments. The default is harpsichord, of course.

   If you use this option, you must specify all instruments. Each $i$ is either an integer between 1 and 255 or a 2-letter mnemonic. The instrument mnemonics allowed with PMX are listed in Table C.1. Numbers and mnemonics may be mixed, but consecutive pairs of numbers must be separated by : (colon)\(^4\). Care is needed with multi-staff instruments: there must be one instrument name per staff; so for a sonata for violin and piano, e.g., the instrument names entry would be $i_{vi}i_{vi}$, not $i_{vi}i_{vi}$!

3. $v[i_1];[i_2];[...];[in]$ assigns the relative MIDI volume to each instrument. Each $i$ is an integer between 1 and 127; the colons are required. The volume parameter $v$ must either be given

\(^4\)Otherwise the assignment could be ambiguous!
with exactly as many parameters as there are instruments, or none at all. The default
(no parameters given) is 127.

4. \texttt{b[m1];[m2];[...];mn]} assigns the MIDI stereo balances to each instrument. The numbers
\textit{m} may vary between 1 and 128; otherwise their usage is similar to that for the volume.
The default value is 64, which represents the center; smaller numbers favor the left stereo
channel, larger ones the right.

5. \texttt{p[x]} inserts a pause of \textit{x} quarter notes – for all instruments! – at the beginning of the
first bar of the input block in which the MIDI symbol appears. Decimals are allowed, but
will be rounded to the nearest sixteenth note.

6. \texttt{g[i]} sets the MIDI gap to \textit{i} MIDI clock tics: this is a silence inserted at the end of every
note, while decreasing the sounding duration by the same amount. The default is 10,
which corresponds to 2/3 of a 64th note.

Key signatures, time signatures (meter) and instrument names will be written into the
MIDI file, the latter as track names. This will have no effect whatsoever on audible output,
but will affect on-screen appearance with some MIDI file players and editors.

The instruments given in Table C.1 are a subset of “The General MIDI Instrument Speci-
fication”. Of course how they sound depends on your hardware and software. Instruments not
listed below can still be used but must be specified by number (cf. Table C.2).

\begin{tabular}{llll}
\texttt{pi} & Acoustic Grand Piano & \texttt{v1} & Violin \\
\texttt{rh} & Rhodes Piano & \texttt{va} & Viola \\
\texttt{ha} & Harpsichord & \texttt{vc} & Cello \\
\texttt{ct} & Clavinet & \texttt{cb} & Contrabass \\
\texttt{or} & Church Organ & \texttt{ab} & Acoustic Bass \\
\texttt{so} & Soprano Sax & \texttt{tr} & Trumpet \\
\texttt{al} & Alto Sax & \texttt{fr} & French Horn \\
\texttt{te} & Tenor Sax & \texttt{tb} & Trombone \\
\texttt{bs} & Baritone Sax & \texttt{tu} & Tuba \\
\end{tabular}

Table C.1: Mnemonics for instruments acceptable in \texttt{PMX}

\section*{C 2.1 MIDI macros} \footnote{\texttt{PMX} normally does not allow tempo changes in MIDI macros. As of version 2.511, this restriction no longer holds; but it is designated as “experimental”. So if you use it, be prepared for surprises! Naturally, Don Simons will certainly be interested to hear of your experiments.}

\texttt{IM} initiates a MIDI macro operation. Although the syntax of a \texttt{PMX} MIDI macro is analogous
to that of regular \texttt{PMX} macros (decribed in Section B 7), its function is different:

1. a MIDI macro cannot be saved (i.e. recorded without playing — there is no command
\texttt{IMS[i]}),
2. the code sandwiched between IMR[‡] (“start record MIDI macro”) and IM (“end MIDI macro”) will be recorded as with a regular macro, but

3. IMP[‡] (“Playback of MIDI macro ‡”) will not insert anything in the written score, it will merely replay in the MIDI output the number of bars defined by the macro.

MIDI macros are needed for repeats or dacapos, as the PMX coding for these features of a score merely provides the appropriate notation on paper, nothing else. MIDI Macros must have ID numbers between 1 and 20. Only one macro may be active at a time, recording or playing, but not both. Never try nesting or overlapping macros!

C 2.2 MIDI only accidentals

In rare cases a special notation for accidentals is needed to guarantee that the MIDI output corresponds exactly to what is intended in the score. Cases in point are:

Repeated notes with accidentals: the generally accepted rules of musical orthography, at least for tonal music, say that when two identical notes with accidentals occur in the same bar (and in the same voice, of course), only the first one is written with an explicit accidental; for the following notes the accidentals are implied (‘inherited’, in PMX parlance).

This rule is less clear when there is a bar line between two successive notes. Some composers – in particular in the baroque era, where this notation is almost the default – will assume the first note in the new bar to have ‘inherited’ the accidental along with it, while others would insist on repeating the accidental, and still others would put a cautionary accidental over the first note of the new bar. PMX takes the first position, and MIDI files generated by PMX will reflect this attitude.

PMX users with a different attitude could thus find their intention misrepresented in the MIDI output.

Editorial/dubious accidentals: in editing an historical piece, a conscientious editor will provide information about dubious points, but she nevertheless, in creating a MIDI file, might want to assert her own position.

For these and similar applications PMX provides the MIDI only accidentals. These are written and used like normal accidentals, except with an ‘i’ added. Such accidentals are effective only in the MIDI output; they are disregarded in the printed score.

This is particularly useful when you want to override the MIDI “baroque default” described above. To understand the “baroque default” clearly, consider the example given in Fig.C.1, where the PMX coding is given directly below the system:

Now compare the printed score with the MIDI version – provided here for your convenience –, and you can hear the differences explicitly.\(^6\)

Another example for the use of MIDI only accidentals is given in Fig.C.2:

\(^6\)It is also instructive to note the MIDI implementation of slurs and ties generated by PMX here (Cf. the list of MIDI restrictions on p.89).
<table>
<thead>
<tr>
<th></th>
<th>PIANO</th>
<th>9-16 CHROMAT. PERC.</th>
<th>17-24 ORGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acoustic Grand Piano</td>
<td>9</td>
<td>Celesta</td>
</tr>
<tr>
<td>2</td>
<td>Bright Acoustic Piano</td>
<td>10</td>
<td>Glockenspiel</td>
</tr>
<tr>
<td>3</td>
<td>Electric Grand Piano</td>
<td>11</td>
<td>Music Box</td>
</tr>
<tr>
<td>4</td>
<td>Honky-tonk Piano</td>
<td>12</td>
<td>Vibraphone</td>
</tr>
<tr>
<td>5</td>
<td>Electric Piano 1</td>
<td>13</td>
<td>Marimba</td>
</tr>
<tr>
<td>6</td>
<td>Electric Piano 2</td>
<td>14</td>
<td>Xylophone</td>
</tr>
<tr>
<td>7</td>
<td>Harpsichord</td>
<td>15</td>
<td>Tubular Bells</td>
</tr>
<tr>
<td>8</td>
<td>Clavinet</td>
<td>16</td>
<td>Dulcimer</td>
</tr>
<tr>
<td>17</td>
<td>Organ</td>
<td>17</td>
<td>Drawbar Organ</td>
</tr>
<tr>
<td>18</td>
<td>Percussive Organ</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Rock Organ</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Church Organ</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Reed Organ</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Accordion</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Harmonica</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Tango Organ</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>GUITAR</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Acoustic Guitar (nylon)</td>
<td>26</td>
<td>Acoustic Bass</td>
</tr>
<tr>
<td>27</td>
<td>Electric Guitar (jazz)</td>
<td>27</td>
<td>Electric Bass (finger)</td>
</tr>
<tr>
<td>28</td>
<td>Electric Guitar (clean)</td>
<td>28</td>
<td>Electric Bass (pick)</td>
</tr>
<tr>
<td>29</td>
<td>Electric Guitar (muted)</td>
<td>29</td>
<td>Fretless Bass</td>
</tr>
<tr>
<td>30</td>
<td>Overdriven Guitar</td>
<td>30</td>
<td>Slap Bass 1</td>
</tr>
<tr>
<td>31</td>
<td>Distortion Guitar</td>
<td>31</td>
<td>Synth Bass 1</td>
</tr>
<tr>
<td>32</td>
<td>Guitar Harmonics</td>
<td>32</td>
<td>Synth Bass 2</td>
</tr>
<tr>
<td>33-40</td>
<td>BASS</td>
<td>33-40</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Acoustic Bass</td>
<td>41</td>
<td>Violin</td>
</tr>
<tr>
<td>42</td>
<td>Viola</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>Cello</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Contrabass</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>Tremelo Strings</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Pizzicato Strings</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>Orchestral Strings</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>Timpani</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>49-56</td>
<td>ENSEMBLE</td>
<td>49-56</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>String Ensemble 1</td>
<td>49</td>
<td>Trumpet</td>
</tr>
<tr>
<td>50</td>
<td>String Ensemble 2</td>
<td>50</td>
<td>Trombone</td>
</tr>
<tr>
<td>51</td>
<td>Synth Strings 1</td>
<td>51</td>
<td>Tuba</td>
</tr>
<tr>
<td>52</td>
<td>Synth Strings 2</td>
<td>52</td>
<td>Muted Trumpet</td>
</tr>
<tr>
<td>53</td>
<td>Choir Aahs</td>
<td>53</td>
<td>French Horn</td>
</tr>
<tr>
<td>54</td>
<td>Voice Oohs</td>
<td>54</td>
<td>Brass Section</td>
</tr>
<tr>
<td>55</td>
<td>Synth Voice</td>
<td>55</td>
<td>Synth Brass 1</td>
</tr>
<tr>
<td>56</td>
<td>Orchestra Hit</td>
<td>56</td>
<td>Synth Brass 2</td>
</tr>
<tr>
<td>57-64</td>
<td>BRASS</td>
<td>57-64</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>Trumpet</td>
<td>57</td>
<td>Soprano Sax</td>
</tr>
<tr>
<td>58</td>
<td>Trombone</td>
<td>58</td>
<td>Alto Sax</td>
</tr>
<tr>
<td>59</td>
<td>Tuba</td>
<td>59</td>
<td>Tenor Sax</td>
</tr>
<tr>
<td>60</td>
<td>Muted Trumpet</td>
<td>60</td>
<td>Baritone Sax</td>
</tr>
<tr>
<td>61</td>
<td>French Horn</td>
<td>61</td>
<td>Oboe</td>
</tr>
<tr>
<td>62</td>
<td>Brass Section</td>
<td>62</td>
<td>English Horn</td>
</tr>
<tr>
<td>63</td>
<td>Synth Brass 1</td>
<td>63</td>
<td>Bassoon</td>
</tr>
<tr>
<td>64</td>
<td>Synth Brass 2</td>
<td>64</td>
<td>Clarinet</td>
</tr>
<tr>
<td>73-80</td>
<td>PIPE</td>
<td>73-80</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>Piccolo</td>
<td>73</td>
<td>Lead 1 (square)</td>
</tr>
<tr>
<td>74</td>
<td>Flute</td>
<td>74</td>
<td>Lead 2 (sawtooth)</td>
</tr>
<tr>
<td>75</td>
<td>Recorder</td>
<td>75</td>
<td>Lead 3 (calliope)</td>
</tr>
<tr>
<td>76</td>
<td>Pan Flute</td>
<td>76</td>
<td>Lead 4 (chiff)</td>
</tr>
<tr>
<td>77</td>
<td>Blown Bottle</td>
<td>77</td>
<td>Lead 5 (charang)</td>
</tr>
<tr>
<td>78</td>
<td>Shakuhachi</td>
<td>78</td>
<td>Lead 6 (voice)</td>
</tr>
<tr>
<td>79</td>
<td>Whistle</td>
<td>79</td>
<td>Lead 7 (fifths)</td>
</tr>
<tr>
<td>80</td>
<td>Ocarina</td>
<td>80</td>
<td>Lead 8 (bass+lead)</td>
</tr>
<tr>
<td>81-88</td>
<td>SYNTH LEAD</td>
<td>81-88</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>Lead 1 (square)</td>
<td>81</td>
<td>Pad 1 (new age)</td>
</tr>
<tr>
<td>82</td>
<td>Lead 2 (sawtooth)</td>
<td>82</td>
<td>Pad 2 (warm)</td>
</tr>
<tr>
<td>83</td>
<td>Lead 3 (calliope)</td>
<td>83</td>
<td>Pad 3 (polysynth)</td>
</tr>
<tr>
<td>84</td>
<td>Lead 5 (charang)</td>
<td>84</td>
<td>Pad 5 (bowed)</td>
</tr>
<tr>
<td>85</td>
<td>Lead 6 (voice)</td>
<td>85</td>
<td>Pad 6 (metallic)</td>
</tr>
<tr>
<td>86</td>
<td>Lead 7 (fifths)</td>
<td>86</td>
<td>Pad 7 (holy)</td>
</tr>
<tr>
<td>87</td>
<td>Lead 8 (bass+lead)</td>
<td>87</td>
<td>Pad 8 (sweep)</td>
</tr>
<tr>
<td>97-104</td>
<td>SYNTH EFFECTS</td>
<td>97-104</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>FX 1 (rain)</td>
<td>97</td>
<td>Sitar</td>
</tr>
<tr>
<td>98</td>
<td>FX 2 (soundtrack)</td>
<td>98</td>
<td>Banjo</td>
</tr>
<tr>
<td>99</td>
<td>FX 3 (crystal)</td>
<td>99</td>
<td>Shamisen</td>
</tr>
<tr>
<td>100</td>
<td>FX 4 (atmosphere)</td>
<td>100</td>
<td>Koto</td>
</tr>
<tr>
<td>101</td>
<td>FX 5 (brightness)</td>
<td>101</td>
<td>Kalimba</td>
</tr>
<tr>
<td>102</td>
<td>FX 6 (goblins)</td>
<td>102</td>
<td>Bagpipe</td>
</tr>
<tr>
<td>103</td>
<td>FX 7 (echoes)</td>
<td>103</td>
<td>Fiddle</td>
</tr>
<tr>
<td>104</td>
<td>FX 8 (sci-fi)</td>
<td>104</td>
<td>Shamel</td>
</tr>
<tr>
<td>105-112</td>
<td>ETHNIC</td>
<td>105-112</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>Tinkle Bell</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>Agogo</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>Steel Drums</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>Woodblock</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>Taiko Drum</td>
<td>109</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>Melodic Drum</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>Synth Drum</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>Reverse Cymbal</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>114</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>115</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>116</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>117</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>118</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>119</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>121</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>122</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>123</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>124</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>126</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>127</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table C.2: The General MIDI Instrument Specification
Figure C.1: The “baroque default”, and overriding it with MIDI only accidentals

Figure C.2: Further use of MIDI only accidentals

- MIDI plays the first C in bar 2 as a C♯. The editorial ♮ over the second C, however, suggests that C♮ was intended (at least from here on), so the MIDI accidental in is appended to that C.

- the final F (in bar 5) suggested to the editor that we are moving towards F major here, and thus the B in bar 3 should actually have been a B♭. By bar 4 the assumption is confirmed, and so he changes the ‘dubious’ accidental to an editorial one, and the MIDI only accidental fi is appended to that B.

C 3 Lyrics

PMX has no special provisions for lyrics. One way to include them is by using the macro package musixlyr.tex developed by Rainer Dunker. It introduces lyrics into TeX more easily than with MusiXTeX’s own facilities. The macros could be entered as inline TeX directly into the .pmx file, but most would prefer the convenient interface to musixlyr via the program M-Tx developed by Dirk Laurie (cf. Section F).

If you have foreign-language lyrics, you may run into the problem of accented letters. Straight TeX does not provide these directly, as the modern font encoding schemes used by LATEX do. However, Olivier Vogel has developed a method with which to use these encoding schemes with M-Tx/PMX/MusiXTeX. For details you should look at his contribution on this in the Tricks and Tricks section in the WIMA, or contact Olivier directly: oliviervogel@freesurf.ch

C 4 PMX and LATEX

LATEX ("A Document Preparation System"), although written for a very different purpose, has a lot in common with PMX: it is a (huge) set of TeX macros that allows to solve complicated layout problems without the need to dive into the – sometimes arcane – depths of TeX itself. So LATEX simplifies the use of TeX for the non-TeXpert in much the same way that PMX does for MusiXTeX.
So it would certainly be wonderful to have the best of two worlds: merge \LaTeX{} and PMX. The problems with that (or merging \LaTeX{} and MusiXTeX, for that matter) are threefold:

1. **PMX** deals with many of the layout aspects of a musical score on its own (such as width of systems, line breaks, page breaks etc.), and with techniques that are not really compatible with those of \LaTeX{}.

2. MusiXTeX / PMX and \LaTeX{} make heavy use of \TeX{}’s resources, both in memory and registers, and together they may strain these resources beyond their usual limit.

3. both MusiXTeX / PMX and \LaTeX{} use many special command definitions, often enough incompatible with one another.

While with modern implementations of resources are no longer a serious problem, the incompatibility problems are, and their resolution would be a major programming task. So there have, to this day, not been any serious efforts to provide a truly merged version of \LaTeX{} with PMX.

There are, however, several methods to use \LaTeX{} and PMX in (partial) “coexistence”:

1. The best way to include short scores, of less than one printed page, in a \LaTeX{} document is to
   
   (a) produce the musical score with \texttt{PMX} – MusiXTeX – \texttt{dvips} in the way described in this tutorial. The end product is a PostScript file,
   
   (b) make an .eps (“encapsulated PostScript”) file from the .ps with one of the standard tools, e.g. \texttt{ghostview},
   
   (c) include this .eps file in the \LaTeX{} document with the \LaTeX{} \texttt{\includegraphics} command, e.g.
   \begin{verbatim}
   \includegraphics[scale=0.58]{sample.eps}
   \end{verbatim}

   This is the method with which the present tutorial was produced.

2. use the \LaTeX{} package \texttt{musixltx}, together with MusiXTeX\texttt{8}. The MusiXTeX code is then sandwiched between the start and end of a specific environment:

   \begin{verbatim}
   \begin{music}
   \end{music}
   \end{verbatim}

   For more detail on this method see the MusiXTeX manual by Daniel Taupin.

   But be aware of the fact that the .tex files produced by PMX contain a number of incompatibilities with \LaTeX{}. So if you want to use the musixltx package together with PMX, you may have to comment out a few lines in the .tex file — by hand, unfortunately. For that reason, the musixltx package usually works well with PMX only in rather simple cases. An example of how to proceed can be found in the WIMA:

   \texttt{http://icking-music-archive.org//software/musixtex/add-ons/pmx-12e.tex}

---

\footnote{In general, this is possible for single-page PostScript files only.}

\footnote{\texttt{musixltx} is part of the standard distribution of the MusiXTeX 1.15 package.}
3. Dirk Laurie, the author of \texttt{M-Tx}, has developed a \LaTeX{} package, \texttt{mtxlatex}. It is part of the \texttt{M-Tx 0.60b distribution}, available from the WIMA, and can be considered a successor to the \texttt{musixltx} package described above. An outstanding feature of this package is that it interfaces quite well with \texttt{PMX}; so this is usually the method of choice in cases where you want to include longer music scores in a long text such as a book. Detailed instructions on how to use this package are contained in the \texttt{M-Tx} manual.

When using \texttt{mtxlatex}, be sure to include the \texttt{mtxlatexit} package as the \textit{first} package in the \LaTeX{} source file.
Chapter D

Limitations, error messages, and bugs

D 1 Limitations

For simplicity in writing the program, PMX has numerous variables with fixed dimensions. In most cases there are no checks against these limits, so occasionally there may be hangups due to exceeding a dimension.

The limits of the variables given in Tables D.1 and D.2 fall in two distinct classes, named “soft” and “hard” limits to distinguish them:

1. soft limits can be increased fairly easily: all it takes (sic!) is to change the dimensions in the FORTRAN source of pmxab.exe (the PMX program) accordingly, and recompile,

2. hard limits, on the other hand, could only be increased with more or less elaborate reprogramming ¹ (or at least recompiling the source code), so for all practical purposes they are unchangeable.

In both cases, however, it will usually be possible to work within existing limits by breaking the input into smaller blocks. Consider that solution before asking Don Simons for changes to the PMX source code, or attempting such changes yourself!

---

¹ Often these hard limits are a consequence of features of MusiXTeX and thus cannot be changed without changes in MusiXTeX.

Note, however, that this character count includes all empty spaces before the final character (such as /) at the end of an input line. If – for better readability, e.g. – you are in the habit of aligning these end-of line characters, being more stingy on the resulting blanc characters often helps to stay within the limit.
D 1. LIMITATIONS

characters per input line : 128
staves : 24
voices per staff : 2
voices per system : 24
systems : 125
bars : 600
forced line breaks : 40
forced page breaks : 10
key changes : 18
pages : 20
notes per input block : 600
bars per input block : 15
slurs per input block : 101
figures (figured bass) per input block : 74
grace note groups per input block : 37
notes in grace note groups per input block : 74
inline \TeX strings per input block : 52
volte per input block : 6
trills per input block : 24
chordal notes (non-spacing) per input block : 62
beams per voice per bar : 8
forced beams per voice per input block : 40
clef changes per voice per input block : 10
notes per beam : 24
notes per xtuplet : 24

Table D.1: Numerical limits of PMX variables (soft limits)

<table>
<thead>
<tr>
<th>Description</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>total number of characters in source file</td>
<td>65 536 (2^{16})</td>
</tr>
<tr>
<td>\texttt{\textbackslash notes} groups (total)</td>
<td>2000</td>
</tr>
<tr>
<td>\texttt{\textbackslash notes} groups per bar</td>
<td>20</td>
</tr>
<tr>
<td>inserted standard anti-collision spaces (not xtuplet or end-of-bar) per bar</td>
<td>20</td>
</tr>
<tr>
<td>inserted anti-collision spaces within xtuplets per bar</td>
<td>20</td>
</tr>
<tr>
<td>inserted anti-collision end-of-bar hardspaces per system</td>
<td>19</td>
</tr>
<tr>
<td>inserted anti-collision end-of-bar hardspaces (total)</td>
<td>83</td>
</tr>
<tr>
<td>inserted standard anti-collision spaces per system</td>
<td>400</td>
</tr>
<tr>
<td>inserted anti-collision spaces within xtuplets per system</td>
<td>100</td>
</tr>
<tr>
<td>inserted standard anti-collision spaces</td>
<td>1000</td>
</tr>
<tr>
<td>inserted anti-collision spaces within xtuplets</td>
<td>200</td>
</tr>
<tr>
<td>bytes of MIDI output data per line of music</td>
<td>24576</td>
</tr>
</tbody>
</table>

Table D.2: Numerical limits of PMX variables (hard limits)
CHAPTER D. LIMITATIONS, ERROR MESSAGES, AND BUGS

D 2 PMX’s error messages

When run on, say, my_opus.pmx, PMX will always generate two files in the working directory, my_opus.pml and pmxaerr.dat. my_opus.pml is a log file, and pmxaerr.dat contains a single integer: 0 if the run was successful, otherwise the line number in the source file my_opus.pmx of the fatal error (useful for batch processing). Also, on successful completion, my_opus.tex will be placed in the path specified in the preamble.

Usually, when there is an error, PMX will give you a pretty good explanation. There are some cases, however, in which the error message will be issued not by PMX, but by the FORTRAN compiler directly, and it may look very cryptic to you. An infamous example is:

```
fortrl: severe (24): end-of-file during read, unit 10
```

Such messages are almost always caused by an extra line in your source file that shouldn’t be there, so experiment with deleting lines towards the end of your file!

D 3 Bugs

As was mentioned before, PMX is continuously maintained and improved by Don Simons, and there is a very active group of PMX users. Thus there is no serious bug known (as of Version 2.514) that hasn’t been fixed.

If you think you have discovered a new bug, don’t hesitate to send a message to the TeX-music users’ list: tex-music@icking-music-archive.org. The same is true if you have a problem with PMX that you simply cannot solve by yourself, even though you suspect that it’s not a bug: don’t hesitate to ask!

D 3.1 A Benign Bug

When \TeX\-ing the output of PMX you will usually get an \texttt{Underfull \vbox} message at the end of each page. This is due to PMX inserting \texttt{\textbackslash eject} at the end of every page, which automatically spaces the systems vertically without having to fiddle with \texttt{\textbackslash staffbotmarg}. As far as is known, the warning is benign, and may be ignored.

---

2Note that sometimes an error message will use a name for a preamble numerical parameter that differs from that used in this tutorial — but it usually is easily recognized.

3Almost all reported “bugs” turn out to be misunderstandings. So if you state politely that you think you may have found a bug, and ask for help, you will avoid irate replies of the “RTFM” species.
Chapter E

Tricks of the Trade

E 1 Simple tricks

E 1.1 Special coding in L’Incoronazione di Poppea

Bar 17 in Fig. B.29 requires an explanation: the continuation figure appears under the 5, but
the these two figures are coded as x145 x12\.200.2. Why not x14500.2, giving them both the
same horizontal offset? That is because the continuation figure always starts 0.3 \noteskip
to the left of the associated note’s position. This works well in most normal situations. But
here \noteskip is large, based on the dotted half, while the desired length is short, so in fact
the left offset of the starting point is even larger than the desired length. Consequently the
entry point of the continuation figure has to be offset by a half note to make it appear offset
by a quarter note.

E 1.2 Text after final system

Suppose you want to add some text, or any other \TeX material, after the final system of a
score. How can you include that in the PMX file?\footnote{This question was posted to the \TeX-music list by Herrmann Hinsch.}

The answer, given by Christian Mondrup, is instructive:

1. Write the text to a separate file, say Myendnotes.tex,

2. Insert

\let\Endpiecesav\Endpiece
\def\Endpiece{\Endpiecesav\input Myendnotes}

in the header of the PMX source file, i.e. as a Type 4 inline \TeX command (cf. Sec. B 8.1). You can see that the redefinition of \TeX commands, if done carefully, can be a rather powerful tool.

Another solution to this problem is to use the \TeX/PMX interface mtxlatex (cf. C 4); this may be simpler in cases involving more complicated layout.
CHAPTER E. TRICKS OF THE TRADE

E 1.3 Clef octaviation

As was noted in Section B 4.14, PMX does not presently provide a notation for octaviation; but native MusiXTeX does; cf. the MusiXTeX 1.15 manual, Sec. 2.11 for details.

Consider, for example, bar 155 of the 2nd movement of Beethoven’s piano sonata op. 111:

\begin{verbatim}
\interstaff{13}
\w170m
Abepl
\def\octnumber{8^{va}}
%  
[ c8d-l \ze+ e8- \ze+ sl \c1- \zc+ sl ] c8d-l \zc+ | Rb /  
[ l c15x3n g+ c b1-x3n g+ \b c1-x3n \g+ c ]  
[ l g1-x3n b gs+ g1-x3n b e a1-x3n e+ a ]  
[ l a-1x3n e+ a g1-x3n e+ g a1-x3n e+ a ] | /  
% 
\end{verbatim}

In most editions, you will find the figures in the right hand in the second and third three beats written in octaviation notation:

\begin{verbatim}
\interstaff{13}
\w170m
Abepl  
\Def{octnumber}{8^{va}}
%  
[ e83d ze+ e8- ze+ sfu c1- \zc+ sfu ] c8d-l \zc+ | Rb /  
[ l c15x3n g+ c b1-x3n g+ \b c1-x3n \g+ c ]  
\Ioctfinup1d \[ l g-s1x3n b gs+ g1-x3n b e a1-x3n e+ a ]  
[ l a-1x3n e+ a g1-x3n e+ g a1-x3n e+ a ] \toctfin1 \ | /  
% 
\end{verbatim}

which is clearly easier to read. This octaviated version was produced by the following PMX code:

\begin{verbatim}
\interstaff{13}
\w170m
Abepl \Def{octnumber}{8^{va}}
%  
[ e83d ze+ e8- ze+ sfu c1- \zc+ sfu ] c8d-l \zc+ | Rb /  
[ l c15x3n g+ c b1-x3n g+ \b c1-x3n \g+ c ]  
\Octfinup1d \[ l g-s1x3n b gs+ g1-x3n b e a1-x3n e+ a ]  
[ l a-1x3n e+ a g1-x3n e+ g a1-x3n e+ a ] \Octfin1 \ | /  
% 
\end{verbatim}

The octaviation is started with the Type 1 inline TeX symbol \Octfinup1d and ended with \Octfin1; the transposition downward is, of course, generated in the standard PMX
way by writing \texttt{g-s1x3n} instead of \texttt{gs1x3n}. By default, MusiXT\TeX\ will start the octaviation symbol with a simple 8; this is changed to \texttt{8^a} by the Type 2 inline \TeX\ symbol given in the preamble.

When using this octaviation notation in a score, do not try to produce a MIDI file for that score: it will come out faulty

E 2 More tricks

The above examples are actually rather simple cases of the use of inline \TeX, not really “tricks” ; they have been included here for you to ‘warm up’ to doing such things yourself. But there are many further-reaching tricks to overcome shortcomings of \texttt{PMX} — some truly ingenious. Here are a few of general interest:

E 2.1 Changing vertical positioning of instrument name

When making parts from a score with \texttt{scor2prt}, you will sometimes find the adjustment of the vertical position of the instrument name not to your liking. In particular, if you have a tempo indicated in a line of text above the first system (e.g. “allegro”), you may find that the instrument name of some (but not all) parts collides with the tempo indication. How can you fix that?

A practical solution of this problem has been given recently by Andre Van Ryckeghem. It is quite instructive to see in this example how the mechanism of handing over commands to \texttt{scor2prt} operates:

Suppose that you are writing a string quartet, and the relevant part of the preamble and header in the score are:

\begin{verbatim}
... Violoncello 
Viola 
Violine II 
Violin I 
batt 
./ 
h 
Allegro 
Abpl 
...
\end{verbatim}

After running \texttt{scor2prt}, you find that the allegro command is positioned the way you want it, but the cello name collides fully, and the Violino II slightly, with the “allegro”. Therefore you want to raise the cello by 3\texttt{ex}, the Violin II by 1.5\texttt{ex}. To do so, replace the above part of the preamble and header in the score by

\begin{verbatim}
... 
%% Violoncello
\end{verbatim}
E 2.2  Xtuplest ending with a rest

As was pointed out in Section B 4.6, PMX does not allow a rest as the last note in an xtuplet.

Can one find a way to circumvent this restriction? A pedestrian, awkward way around might be this: remembering that PMX is a preprocessor for MusiXTeX, and in a second pass...
\LaTeX{} processes the \texttt{PMX} output, one could simply enter some arbitrary note in place of the desired rest, run \texttt{PMX}, and then in the resulting \LaTeX{} file replace the arbitrary note \textit{manually} by the appropriate \texttt{MusiXTeX} rest symbol \textit{before} running the file through \LaTeX{} for the the 2nd pass.

This semi-solution however, has two major drawbacks:

1. it takes a certain amount of acquaintance with basic \texttt{MusiXTeX}, so this is not for everyone;

2. more dangerously, this hack goes away if and when you rerun \texttt{PMX}! So if you are still in the middle of a development, you may find yourself redoing the hack over and over, which could quickly become a real nuisance.

Luckily, for xtuplets with stand-alone notes there now exists a real solution: an ingenious, generally usable inline \LaTeX{} method – developed by Don Simons – solves the problem completely for unbeamed xtuplets. Here is a simple example with two such triplets:
And this is the PMX source text that creates these staves:

\begin{verbatim}
2 2
2 4 0 0
0 0
0 3 20 0.12

\def\qpforqu{\let\qut\qu\def\qu##1{\qp\zcharnote{##1}{~}\let\qu\qut}}
\def\dsforcu{\let\cut\cu\def\cu##1{\ds\zcharnote{##1}{~}\let\cu\cut}}
e4ax3 \dsforcu\ e \dsforcu\ e e4ax3 \qpforqu\ eD /
\end{verbatim}

As you can see, there are two inline commands of type 2, which are therefore effective throughout the file (from where they are first entered), then followed by two standard PMX lines; the lower one (the top staff) of these has the standard triplets, as demonstration, while the upper one (bottom staff) shows the corresponding triplet s with ending rests.

The two inline \TeX commands produce the type of rest: \dsforcu stands for a quarter rest, and \qpforqu for a half rest. Note that these inline \TeX commands not only provide the correct graphics, but PMX considers them to have the corresponding duration values! So if you use wrong ones, PMX will complain in the usual way.

Explaining the inline \TeX commands in detail would go beyond the scope of this tutorial. But if you know some of the MusiXTeX terminology, you will recognize that these commands conform to the MusiXTeX names for quarter and half notes etc.\footnote{This knowledge may be useful if you want to produce more complicated xtuplets with ending rests than the simple examples given here. The new MusiXTeX 1.15 manual may be helpful in such cases. If that doesn’t help: consult a MusiXTeXpert!}

The inline commands are then each followed by a dummy note. Usually it is a good (and safe) habit to use the same note name as the last real note in the xtuplet; but any PMX note name is valid; variations can produce surprising, but possibly useful graphical results. So if you feel up to it, you can play with such variations!

\section*{E 2.3 Shorthand notation for consecutive quavers}

Often, in accompanying voices, in particular, there are several repeated quavers (eighth notes), e.g. 4 quavers to a half note. There is a commonly used shorthand notation for this that helps sight-reading enormously: a half-note, with a line through its stem, indicating the quavers.

Fig. E.1 shows the beginning of the cello part of a Joh. Chr. Bach quartet. In this example an inline MusiXTeX macro (devised by Andre Van Ryckeghem) provides this shorthand.
If you want such a shorthand for semiquavers (sixteenth notes), you need to consult the MusiXTeX 1.15 manual: you will find that all you need to do is to replace \ibl{0} and \ibl{0} by \ibbl{0} and \ibbl{0}, resp.

E 2.4 Varying the stave sizes

In modern editions of works with basso continuo e.g., the implementation of the bass, as suggested by the editor, is often given in a staff using a smaller size. MusiXTeX does not support such a possibility.

Fortunately, Mthimkhulu Molekwa has written two macros that provide alternative solutions of this problem that are often sufficient. The first, musixbar.tex, which is included in
MusiXTeX, allows you to easily define a set of staves (the two bottom ones, in the case of a basso continuo) that have common bar lines joining these staves and only these.

The second macro, named curly.tex, provides a brace as an accolade of a set of staves together.

Suppose we want to set a sonata for violin and basso continuo with PMX. There will be three staves: at the bottom the basso continuo proper, then immediately above it, and grouped together with a brace accolade (in a smaller size), its implementation, and above these the violin staff.

To get this, do the following:

- indicate in the preamble the number of instruments, three in this case!
- add a Type 4) \TeX inline command at the beginning of the file, invoking the macros musixbar.tex and curly.tex, e.g.:

```
\let\:=\relax\input musixtex\:sepbarrules\input pmx
\input musixbar\input curly
```

- insert the following \TeX inline commands in the header of the PMX input file:

```
\\\indivbarrules\sepbarrule3\n\setsize2\smallvalue\curlybrackets{{1}{2}}\n\let\interstaffsav\interstaff\def\interstaff#1{}
\interstaffsav{9}
```

The command \interstaffsav{9}\ determines the extension of the brace (and can be adjusted). The command \indivbarrules\sepbarrule3\ says that the third staff, that for the violin, has its own discontinuous bar line, not joined with the other two.

The second line contains two commands: \setsize2\smallvalue sets the size of staff 2 to be smaller, and \curlybrackets{{1}{2}}\ says which staves are to be accoladed, i.e. joined by the brace.

The file curly.tex must, of course, be in a directory where PMX and \TeX can find it, viz. either in the current directory or in one in which the other MusiX\TeX files are stored.

---

3 Normally, MusiX\TeX will draw common bar lines for all staves or for none.
4 curly.tex, if not included in your MusiX\TeX installation, is available in the “add-ons” software section of the WIMA.
5 MusiX\TeX provides only the choir-type square accolades. For the practical usage of curly.tex see the Caccini example in the appendix, Fig. G.3.
6 The MusiX\TeX command \sepbarrules\ sets the vertical bar to be discontinuous, i.e. not extend across different staves.
E 2.5 Stuff in front of the clefs of the first system

And finally, to show you what is possible, here is a special example of using inline \TeXX: it was contributed to the MusiX\TeXX users' list by Olivier Vogel. Here is the literal text of his contribution, with the result shown in Fig. E.2:

Dear all,

I’ve got a score of “Locus iste” by Bruckner, which begins with the indication of the range of voices. I tried to reproduce the result, and share with you my solution, since it seems to me to be an interesting trick.

---
\input musixtex
\input musixlyr
\setlyrics{soprano}{\%
Lo-cus i-ste a De-o fa-ctus est lo-cus i-ste a De-o \\
fa-ctus est, a De-o, De-o fa-ctus est in-ae-sti-ma-bi-le}
\copylyrics{soprano}{alto}
\copylyrics{soprano}{tenor}
\copylyrics{soprano}{basse}
---

4 4 4 4 0 6 0 0 1 2 20 0.04

bttt
./
B
Abd
h180m
%1=2
\staffbotmarg3\Interligne
\setclefsymbol{2}{\treblelowoct}
\groupbottom{1}{1}\grouptop{1}{4}
\sepbarrules
\startmuflex\indent\hskip-\parindent\hbox{\vbox{\hsize=\parindent\n
\setclefsymbol{1}{\empty}\setclefsymbol{2}{\empty}\n
\setclefsymbol{3}{\empty}\setclefsymbol{4}{\empty}\n
\grouptop{1}{0}\nostartrule\generalmeter{}\parindent=0pt\n
\startpiece\hardspace{2pt}\notes\zq{F}\nq{''C}&\zq{c}\nq{''e}&\n
\zq{N}\nq{''b}&\zq{b}\nq{''g}\en\zstoppiece}}\n
\assignlyrics{1}{basse}\assignlyrics{2}{tenor}\n
\assignlyrics{3}{alto}\assignlyrics{4}{soprano}\n
...
If you encounter a tricky PMX problem that you think is of general interest, so that it may have been solved by someone else already, a good place to search is the PMX ‘Tips and Tricks’ section of the WIMA.

And if you have found a nice trick yourself, don’t hesitate to share it via the TeX-music users’ list. Perhaps it can be included in ‘Tips and Tricks’!
Chapter F

An Extension of PMX: M-Tx

Although PMX is already vastly simpler to use than MusiXTpX, anything can be improved. So Dirk Laurie set out to simplify PMX even further, and thus created M-Tx.

M-Tx is actually a preprocessor to PMX. Its input is a file with extension .mtx, for example mymadrigal.mtx, its output has the extension .pmx, mymadrigal.pmx. Its input language is similar, but not identical, to that of PMX and includes most of the functionality of PMX as a subset.

The major purpose of M-Tx is to facilitate the introduction of lyrics in a musical score of a song, cantata or opera. M-Tx does this in conjunction with the musixlyr package by Rainer Dunker; so if you want to use M-Tx, you ought to have musixlyr installed as well.

Laurie’s Work on M-Tx had essentially been ended in November 1998 with the ‘final’ version 0.52. But, as many such projects, M-Tx continued to evolve; presently, the ‘official’ version is 0.60, dated March 2005, and the current patch is 0.60c, dated 22 November 2007!

Since there is a full manual on M-Tx available in WIMA, these few remarks may suffice here.
Chapter G

Appendix: Examples

G 1 Dons Example Files

The ‘official’ distribution files for PMX, which are available from WIMA [(pmx2514.zip)], contain 3 instructive examples (not reproduced here):

- **most.pmx** contains examples of most of the PMX commands, and a few programming tricks, including examples in the last line of beam groups whose notes vary widely in pitch. The printed output displays the PMX commands near to the resulting typeset characters. It is more useful to look at the printed output rather than the source file, since the file is littered with Inline TeX needed to output the text strings representing the PMX commands. **WARNING:** Do not try to play this music; it could be hazardous.

- **barsant.pmx** contains the first movement of a recorder sonata by the Italian Francesco Barsanti (1690-1772). It demonstrates many of PMX’s strong points in a ‘battlefield situation’: figured bass, complex beaming patterns, xtuplets, and automatically adjusted horizontal and vertical spacing in crowded scores. In fact, this single-page score is at the limit of vertical crowding. It uses the global option *Ae* for equal space between systems. The space between systems was increased (using the option *A11.1*) to give a more pleasing appearance. This is a good score to try making parts with **scor2prt**. The special command %2S9 is used to increase the number of systems in the recorder part (as explained in Section C 1.2).

- **mwalmd.pmx** is an Allemand for harpsichord by the German Matthias Weckmann (1616-1674). It uses many techniques peculiar to keyboard scores.

G 2 Full-score examples

In this appendix you will find the PMX code (by Luigi Cataldi) of a full piece, together with the actual score which was generated from it (Section G 2.1), as well as two extended M-Tx examples (also coded by Luigi Cataldi).
The **PMX** code for the Dufay Kyrie contains a few inline **TeX** commands; this is done intentionally so you again can get a feel of the usage of inline **TeX**; to understand them fully, however, you will have to consult the MusiX**TeX** 1.15 documentation. Similarly, you can get an idea from the Vivaldi and Caccini example what **M-Tx** is all about. For a full understanding, you again need to look at the **M-Tx** documentation.

G 2.1 Dufay, *Kyrie* (PMX code):

```latex
\%----------------%
\%
\% Dufay, Kyrie
\%
\%----------------%
\%
---
\font\tit = cmcsc10 scaled \magstep 5
\font\dat = cmr12
\def\comp{\rightline{\medtype Guillaume Dufay}}
\def\data{\rightline{\medtype (1400?--1474)}}
---
2 1 3 4 3 4 0 0
1 6 20 0

bt
./
\def\writebarno{\ifnum\barno>1\lrlap{\oldstyle\the\barno\barnoadd}\fi}%
\def\shiftbarno{0\Interligne}\
Tt
{\tit Kyrie}
Tc
{\vbox{\comp\data}

It92iororb60:68
\%
1-6
h-4

Kyrie
da23 | a23 bf4 | c24 d4 | ad23 | dd24 | a23 d44 //
a23 d4 | c2 d4 | e2 d4 | csd2 | f2 f4 | e2 f4 /
a24 a4 | e2 d4 | g2 f4 | e2 r4 | ad4 g8 a b | cd45 a84 b c85 /

\%
7-9
c24 bf43 | ad2 | g2 d4 //
e24 d4 | e d8 c4 bf8 | d4 r7 d /
a84 bf a g f | ad2 | bf2 a4 /

\%
10-12
e2 d4 | g4 bf a | gd2 Rd //
g8- f g e f4 | r8+8 [ d+ d cs c bn ] | dd2 /
g2 a4 | g2 fs4 | gd2 /
\% 13-18
L3Mc+4
h=5
Christe
dd23 | fd2 | gd2 | d2 d4 | e2 d4 | g2 a4 //
dd24 | c2 f8 e | d4 c bf | ad2 | g2 f4 | r8+7 [ d+ c b c d ] /
ad24 s | ad2 s | r8 [ b f b a b g ] | fsd2 | g2 a4 | d- e f /

\% 19-24
bf23 a4 | gd2 | d24 c4 | bf43 a g | d f e | dd2 Rd //
g4-r d+ c | d2 r4 | f2 e4 | d c b | a2 gs4 | ad2 /
g24 f4 | g r g | a bf a | g8 f4 e8 g d | f e c d cs b | dd2 /

\% 25-28
L5Mc+4
h=5
Kyrie
d23 g4 | d r8 d e f | g4 f ef | d r0 r0 //
d24 ef4 | d2 c4 | bf2 c4 | dd2 /
a24 g4 | f2 g4 | bf a g | fsd2 /

\% 29-32
g2 a4 | g2 f4 | g2 a4 | d2- d4 //
d2 cs4 | d2 a4 | bf2 c4 | ad2 /
r8 [ g g fs f e ] | g4 f8 g a bf | g4 f e | d r r /

\% 33-36
g43 f g | a f e | d d+ c | d f e //
bf4 a g | fr a8 g4 f8 | a4 bf8r g4 a8 | d4-r d8+r c4 bf8 /
e84 d4 c8 bf4 | a8 c4 d8 c4 | d8 f4 g8 e4 | f8 d4 a8+ g4 /

\% 37-39
d2 c4 | dd2 | gd2- //
d2 e4 | fd2 of-2 | dd2 of-2 /
a4 b c | ad2 of | gd2 of /
\%
\%------------------\%
\%
\% end of Dufay, Kyrie
\%
\%------------------\%
Figure G.1: G. Dufay, *Kyrie* (generated by PMX)
G 2.2 Vivaldi, *Mundi Rector* (M-Tx code):

\%
\%
\%----------------%  
\%
\% Vivaldi, Mundi Rector  
\%
\%----------------%  
\%
Title: \vbox{\titA\titB}  
Composer: \vbox{\comp\data}  
Flats: 2  
Meter: 3/8  
Pages: 1  
Systems: 2  
Style: SATB4  
Size: 16  
Space: 6 6 6 12
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
MUNDI RECTOR DE CAELO MICANTI
& IN JUDITA TUAE LEGI DICATA
& REDI, REDI IAM VICTRIX PUGNANDO \\

AUDI PRECES ET SUSCIPE VOTA
& FLAMMAS DULCIS TUI AMORIS ACCENDE
& IN CILICIO IN PRECE RIVIVE \\

QUAE DE CORDE PRO TE DIMICANTI
& FERITATIS SIC HOSTIS DOMATA
& DE HOLOFERNE SIC HODIE TRIUMPHANDO \\

SUNT PIETATIS IN SINU DEVOTA.
& IN BETHULIAE SPEM PACIS INTENDE.
& PIA JUDITHA PER SAECULA VIVE. \\

% end of Vivaldi, Mundi Rector
%
Mundi rector
Juditha Triumphans, I, 27
Antonio Vivaldi
(1678-1741)

Allegro

Mundi Rector de Caelo mi-canti
Audi preces, audi preces, et susci-pe
vo - ta
In Ju-di-ta teue le - gi di - cata
Flammae dulcis, flammae dulcis, tu - i amorisac-cen - de
Redi, re-di iam Vietrix pugnando
In ci - li-cio, in ci - li-cio in prece ri - vi - ve

Mundi Rector de Caelo mi-canti
Audi preces, audi preces, et susci-pe
vo - ta
In Ju-di-ta teue le - gi di - cata
Flammae dulcis, flammae dulcis, tu - i amorisac-cen - de
Redi, re-di iam Vietrix pugnando
In ci - li-cio, in ci - li-cio in prece ri - vi - ve

Mundi Rector de Caelo mi-canti
Audi preces, audi preces, et susci-pe
vo - ta
In Ju-di-ta teue le - gi di - cata
Flammae dulcis, flammae dulcis, tu - i amorisac-cen - de
Redi, re-di iam Vietrix pugnando
In ci - li-cio, in ci - li-cio in prece ri - vi - ve

Quae de cor - de pro te di-mi - can - ti Sunt pie - ta-tis in si-nu de - vo - ta.
Fe - ri - ta - tis sic hostis do - ma - ta In Be - thliae spem pacis in - ten - de.
De Ho - lo - fer - ne sic hodie triumphando Pi - a Ju-di-tha per saecu - la vi - ve.

Quae de cor - de pro te di-mi - can - ti Sunt pie - ta-tis in si-nu de - vo - ta.
Fe - ri - ta - tis sic hostis do - ma - ta In Be - thliae spem pacis in - ten - de.
De Ho - lo - fer - ne sic hodie triumph - do Pi - a Ju-d - thy a per saecu - la vi - ve.

Quae de cor - de pro te di-mi - can - ti Sunt pie - ta-tis in si-nu de - vo - ta.
Fe - ri - ta - tis sic hostis do - ma - ta In Be - thliae spem pacis in - ten - de.
De Ho - lo - fer - ne sic hodie triumph - do Pi - a Ju - di-tha per saecu - la vi - ve.

Quae de cor - de pro te di-mi - can - ti Sunt pie - ta-tis in si-nu de - vo - ta.
Fe - ri - ta - tis sic hostis do - ma - ta In Be - thliae spem pacis in - ten - de.
De Ho - lo - fer - ne sic hodie triumph - do Pi - a Ju - di-tha per saecu - la vi - ve.

Mundi Rector de Caelo micanti In Juditia tuae lepici dicitata
Audi preces et susci pe vota Flammae dulcis tui amoris acceande
Quae de corde pro te dimicanti Feritas si -c hostis domata De Holoferne sic hodie triumphando
Sunt pietatis in sinu devota. In Bethuliae spem pacis intende. Pia Juditha per saecula vive.

Figure G.2: A. Vivaldi, Mundi Rector (generated by M-Tx/PMX)
G 2.3  Caccini, *Amor l'ali m'impenna* (M-Tx code)

Title: {\tit {Amor l'ali m'impenna}}
Composer: \vbox{\compA\compB}
Meter: C
Style: Singer Cont Bass
Singer: Voices Sop; Vocal; Clefs G
Cont: Voices RH1,RH2; Clefs G
Bass: Voices B; Clefs F
Flats: 1
Systems: 12
Pages: 3
Space: 4 1
Size: 20
Indent: 0.10
Name: {\it{Soprano}} ~ {\it{Continuo}}

\%\input musixbar\input curly
\%\font\data = cmr10
\%\font\tit = cmb10 scaled \magstep 4
\%\font\rxii = cmr12
\%\font\mov = cmmib10 scaled \magstep 1
\%\font\num = cmr9
\%\def\compA{\rightline{\rxii Giulio Caccini}}
\%\def\compB{\rightline{\data (1550--1618)}}
\%\def\writebarno{\ifnum\barno>1\lrlap{\oldstyle\the\barno\barnoadd}\fi}%
\%\def\shiftbarno{0\Interligne}

\{Aria\}
A-mor l'a-li m'im-pen-na.
A-mor dol-ce, a-mor ca-ro, _a-mor fe-li-ce.
Tal che non spe-ro pi{\u} n\{\e\} pi{\u} mi li-ce.
Pas-so nem-bi_e pro-cel-le,
pas-so'l ciel e le stel-le,
del pia-cer que-st'{\e}'l re-gno.
Ah, mia for-tu-na non se l'ab-bia a sde-gno.
Que-sto, que-sto m'ac-co-ra:
ch'al-tri ca-deo,
ch'al-tri ca-deo dal pa-ra-di-so_an-co-ra.
Ah, mia for-tu-na non se l'ab-bia a sde-gno.
Que-sto, que-sto m'ac-co-ra:
ch'al-tri ca-deo,
ch'al-tri ca-deo dal pa-ra-di-so_an-co-ra.
% Bar 8
g0
d0 zg-
ce2 bn
\zcharnote{-6}{\num 11}\ ( g2 \zcharnote{-6}{\num 10}\ g )

% Bar 9
g0
e0+
c0 zg
c0

% Bar 10
c4 g8 g8 b4 [ a1 b1 c1 b1 ]
g2 g
c2 ze d zb
c2 g2 -

% Bar 11
ad4 d8- f4 [ e1 f1 g1 f1 ]
f2.g
a2 zd d zb
d2+ - b4 g4 -

% Bar 12
e0
\zw e\ dr2 cs
a2d-2 g4
%%%%% The two following lines must actually be written in one line!
\zcharnote{-6}{\num 11}\ (5t a2 \zcharnote{-6}{\num $\sharp$10}\ (6t a4 )5t a 7 )6t

% Bar 13
d2 r4 fs8 f8
d2 fs
fs2 zd+ d za
d2 # d2

% Bar 14
g4 d8 d8 (~ [ e8 f1 e1 ] [ f1 e1 d1 e1 ] ~
g4 fn e2
b2 zd c zg
g2- c2

% Bar 15
f2 r4 fs8 f8
f2 fs
a2 zc d za
f2 d2
G 2. FULL-SCORE EXAMPLES

% Bar 16
g4 d8 d8 [ e8 f8 fd8 e1 ]
g4 fn e2
b2 zd c zg
g2- c2

% Bar 17
f4 f8 g8 a2
f2 f
a2 zc c za
f2 f4 e8 d8

% Bar 18
cd8 c1 (~ [ b1 a1 g1 f1 ] g2 )~
e4 ( f f ) e
g2 zc c zg
%%% The two following lines must actually be written in one line!
( c4 \zcharnote{-6}{\num 11}\ c4 )
\zcharnote{-6}{\num 11}\ c4 \zcharnote{-6}{\num 10}\ c4

% Bar 19
f0
f0
a0 zc
f0-

% Bar 20
ad2 g8 f8
f4 ( c c ) bn
a4 g ( f f )
f4+ e4 ( d4 d )

% Bar 21
e8 d8 e8 f8 g2
c2 r8+0 e d c
g2 ze rb
( c2 c8 ) c8 b8 a8

% Bar 22
g4 d4 d2
bnr4 ( c c ) b
g0 zd+
%%% The two following lines must actually be written in one line!
\zcharnote{-6}{\num $\sharp$10}\ ( g4 \zcharnote{-6}{\num 11}\ g )
( \zcharnote{-6}{\num 11}\ g4 \zcharnote{-6}{\num $\sharp$10}\ g )

% Bar 23
c0
e0
c0 zg
c0

% Bar 24
gd4+ e8 a4 g8 f8
e2 e4 f
g2 zc c za
c2 a4 f4

% Bar 25
g2 f8 f8 f8 e8
e2 f
g2 zc c za
c2+ f2-

% Bar 26
d2 r8 b8+ b a
f2 g
b2 zd d zb
b4 b8 a8 g2 -

% Bar 27
g8 e8 e8 f8 g4 [ f1 g1 a1 b1 ]
g4 f e ( f
c2 ze c zg
c2 c4 \zcharnote{-4}{\num 11}\ c4

% Bar 28
g0
f2 ) cr4 b
g2 zc e zg-

%%%%%% The two following lines must actually be written in one line!
\zcharnote{-4}{\num 11}\ (5t c2
(6t \zcharnote{-4}{\num 10}\ c4 )5t c 7 )6t

% Bar 29
f0
f0+
a0
f0-

% Bar 30
ad2 g8 f8
f4 ( c c ) bn
a4 g ( f f )
f4+ e4 ( d 7 d #6 )

% Bar 31
e8 d8 e8 f8 g2
c2 r8+0 e d c
\textbf{G 2. FULL-SCORE EXAMPLES}

\begin{verbatim}
\input{example1}
\end{verbatim}
% Bar 40
[ g1 c1- d1 e1 ] [ f1 g1 a1 b1 ] [ c1 b1 a g1 ] [ a1 g1 a1 f1 ]
e0
c0 zg
c0

% Bar 41
g0 )~
f2 e
\zw N\ ce2d+0+1 b4
%%%%%% The two following lines must actually be written in one line!
\zcharnote{-4}{\num 11}\n(1t c2 (2t \zcharnote{-4}{\num 10}\ c4 )it c 7 )2t

% Bar 42
f0
f0
a0
f0-
%
%-----------------
%
% end Caccini, Amor l’ali m’impenna
%
%-----------------
Amor l’ali m’impenna

Giulio Caccini
(1550–1618)
Ah, mia fortuna non s'abbia a sdegnio.

Questo, questo m'accore: ch'altri cadeo, ch'altri cadeo dal paradiso ancora.

Figure G.3: G. Caccini, *Amor l'ali m'impenna* (generated by M-Tx/PMX)