MusicTEX

Using TEX to write polyphonic or instrumental music

Version 5.04 – August 30, 1994

Daniel TAUPIN
Laboratoire de Physique des Solides
(associé au CNRS)
bâtiment 510, Centre Universitaire, F-91405 ORSAY Cedex

Contents

Chapter 1

What is MusicTEX ?

MusicTEX is a set of TEX macros to typeset polyphonic, orchestral or polyphonic music. Therefore, it is mainly supposed to be used to type wide scores – just because true musicians seldom like to have to frequently turn pages – and this is not really compatible with LATEX's standard page formats, even the `A4.sty` the `\textheight` and `\textwidth` of which are too small for musician needs.

However, a LATEX style has been also provided (and it is used for the typing of the present paper) but this `musictex` style is fit for musicographic books rather than for normal scores to be actually played.

It should be emphasized that MusicTEX is not intended to be a compiler which would translate some standard musical notations into TEX nor to decide by itself about aesthetic problems in music typing. MusicTEX only typesets staves, notes, chords, beams, slurs and ornaments as requested by the composer. Since it makes very few typesetting decisions, MusicTEX appears to be a versatile and rather powerful tool. However, due to the important amount of information to be provided to the typesetting process, coding MusicTEX might appear to be awfully complicated, just as the real keyboard or orchestral music. Therefore, it should be interfaced therefore by some pre-compiler in the case of the composer/typesetter wanting aesthetic decisions to be automatically made by somebody (or something) else.

1.1 MusicTEX principal features

1.1.1 Music typesetting is two-dimensional

Most of the people who just learnt a bit of music at college probably think that music is a linear sequence of symbols, just as literary texts to be TEX-ed. In fact, with the exception of strictly monodic instruments like most orchestral wind instruments and solo voices, one should be aware that reading music actually is a matricial operation: the non-soloist musician successively reads *columns* of simultaneous notes which he actually plays if he is a pianist, clavichordist or organist, which he actually reads and watches if he conducts an orchestra, and which he is supposed to check and partially play when he is a soloist who wants to play in time with the accompanying instrument or choir.

In fact, our personal experience of playing piano and organ as well as sometimes helping as an alternate Kapellmeister leads us to think that actual music reading and composing is a slightly more complicated intellectual process: music reading, music composing and music thinking seems to be a three layer process. The musician usually reads or thinks several consec-

note sequence one	*note seq. four*	*note seq. seven*	*note seq. ten*
note sequence two	*note seq. five*	*note seq. eight*	*note seq. eleven*
note seqence three	*note seq. six*	*note seq. nine*	*note seq. twelve*

Table 1.1: The order in which a musician reads music

utive notes (typically a long *beat* or a group of logically connected notes), then he goes down to the next instrument or voice and finally assembles the whole to build a part of music lasting roughly a few seconds. Then he handles the next *beat* or *bar* of his score.

Thus, it appears that the humanly logical way of coding music consists in horizontally accumulating a set of *vertical combs* with *horizontal teeth* as described in Table 1.1.

This is the reason why, in **MusicTEX** the fundamental *macro* is of the form

```
\notes ... & ... & ... \enotes
```

where the character `&` is used to separate the notes to be typeset on respective staffs of the various instruments, starting from the bottom.

In the case of an instrument whose score has to be written with several staffs, these staffs are separated by the character `|`. Thus, a score written for a keyboard instrument and a monodic instrument (for example piano and violin) will be coded as follows:

```
\notes ... | ... & ... \enotes
```

for each column of simultaneous *groups of notes*. It is worth emphasizing that we actually said *"groups of notes"*: this means that in each section of the previous macro, the music typesetter is welcome to insert, not only chord notes to be played at once, but small sequences of consecutive notes which build something he understands as a musical phrase. This is why note typing macros are of two kinds in MusicTEX, namely the note macros which are not followed by spacing afterwards, and those which induce horizontal spacing afterwards.

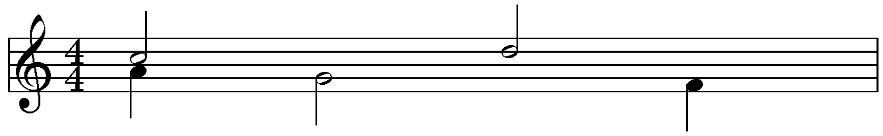
1.1.2 The spacing of the notes

It seems that many books have dealt with this problem. Although it can lead to interesting algorithms, we think it is not in practice an important point.

In fact, each column of notes has not necessarily the same spacing and, in principle, this *spacing* should depend on the shortest duration of the simultaneous notes. But this cannot be established as a rule, for at least two reasons:

1. spacing does not depend only on the local notes, but also on the context, at least in the same bar.

2. in the case of polyphonic music, exceptions can easily be found. Here is an example:



where it can be clearly seen that the half notes at beats 2 and 3 must be spaced as if they were quarter notes since they overlap, which is obvious only because of the presence of the indication of the *meter* 4/4.

Therefore, we preferred to provide the composer/typesetter with a set of macros, the spacing of which increases by a factor of $\sqrt{2}$ (incidentally, this can be adjusted):

```
\notes ... & ... & ... \enotes  % 1 spatial unit
\Notes ... & ... & ... \enotes  % 1.4 spacial unit
\NOtes ... & ... & ... \enotes  % 2 spatial units
\NOTes ... & ... & ... \enotes  % 2.8 spatial units
\NOTEs ... & ... & ... \enotes  % 4 spatial units
\NOTES ... & ... & ... \enotes  % 5.6 spatial units
```

The size of the spatial unit (`\elemskip`) can be freely adjusted. In addition, MusicTEX provides a means of adjusting the note spacing according to an average number of elementary spaces within a line (macro `\autolines`).

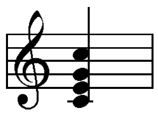
1.1.3 Music tokens, rather than a readymade generator

The tokens provided by MusicTEX are:

- the note symbols *without stem*;
- the note symbols *with stems, and hooks for eighth notes and beyond*;
- the indications of beam beginnings and beam ends;
- the indications of beginnings and ends of ties and slurs;
- the indications of accidentals;
- the ornaments: arpeggios, trills, mordents, pincés, turns, staccatos and pizzicatos, fermatas;
- the bars, the meter and signature changes, etc.

Thus, `\wh a` produces an *A (of nominal frequency 222.5 Hz, unless transposed)* of duration being a *whole note*. In the same way, `\wh h` produces an *A (445 Hz)* of duration represented by a *whole note*, `\qu c` produces a *C (250 Hz approx.)* whose value is a *quarter note with stem up*, `\cl J` produces a *C (125 Hz approx.)* of duration equal to an *eighth note with stem down*, etc.

To generate quarter, eighth, sixteenth, etc. chords, the macro `\zq` can be used: it produces a quarter note head the vertical position of which is memorized and recalled when another stemmed note (possibly with a hook) is coded; then the stem is adjusted to link all simultaneous notes. Thus, the perfect C-major chord, i.e.



is coded `\zq c\zq e\zq g\qu j` or, in a more concise way, `\zq{ceg}\qu j` (stem up): in fact, single notes are treated... like one-note chords.

1.1.4 Beams

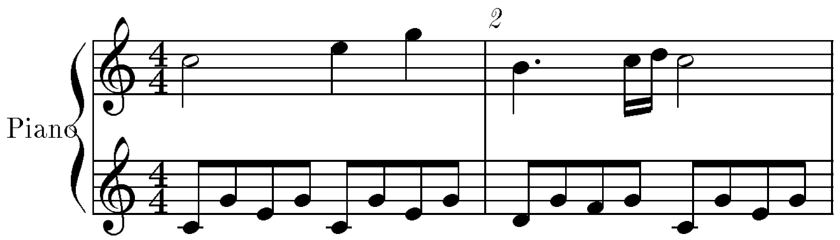
Beams are generated using macros which define their beginning (at the current horizontal position), together with their altitude, their direction (upper of lower), their multiplicity, their slope and their reference number. This latter feature – the reference number – appears to be necessary, since one may want to write beams whose horizontal extents overlap: therefore, it is necessary to specify which beam the notes hang on and which beam is terminated at a given position.

1.1.5 Setting anything on the score

A general macro (`\zcharnote`) provides a means of putting any sequence of symbols (in fact, some `\hbox{...}`) at any pitch of any staff of any instrument. Thus, any symbol defined in a font (letters, math symbols, etc.) can be used to typeset music.

1.2 A simple example

Before giving more details, we give below an example of the two first bars of the sonata in C-major K545 by MOZART:



The *coding* is set as follows:

```
\begin{music}
\parindent 1cm
\def\nbinstruments{1}\relax % a single instrument
\def\instrumenti{Piano}%    % whose name is Piano
\nbporteesi=2\relax         % with two staffs
\generalmeter{\meterfrac{4}{4}}\relax % 4/4 meter chosen
\debutextrait               % starting real score
\normal                     % normal 12 pt note spacing
\temps\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\hl j\enotes
\temps\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\ql l\sk\ql n\enotes
\barre                      % bar
\Notes\ibu0f0\qh0{dgf}|\qlp i\enotes
\notes\tbu0\qh0g|\ibbl1j3\qb1j\tbl1\qb1k\enotes
\temps\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\hl j\enotes
\finextrait                 % terminate excerpt
\end{music}
```

- `\ibu0f0` begins an upper beam, aligned on the f, reference number 0, slope 0
- `\tbu0` terminates this beam before writing the second g by means of `\qh0g`

- `\qh..` indicates a note hanging on a beam.
- `\sk` sets a space between the two quarters at the right hand, so that the second is aligned with the third eighth of the left hand.
- `\qlp` is a quarter with a point.
- `\ibbl1j3` begins a double beam, aligned on the C (`j` at this pitch) of slope 0.15.

1.3 Some highlights

1.3.1 Signatures

Signatures can be stated either for all instruments, for example by `\generalsignature{-2}` which sets two flats on each staff, or separately for each instrument. Thus, the `\generalsignature` can be partly overridden by `\signii=1` which puts one sharp on the staffs of *instrument number* 2 (ii). Of course, the current signature may change at any time as well as the meters and clefs.

1.3.2 Transposition

An important question is: *"can MusicTEX transpose a score ?"*. The answer is now 99.5 % *yes*. If fact, there is an internal register named `\transpose` the default value of which is zero, but it may be set to any reasonable positive of negative value. In that case, it offsets all symbols pitched with letter symbols by that number of pitch steps. However, it will neither change the signature nor the local accidentals, and if – for example – you transpose a piece written in C by 1 pitch, MusicTEX will not know whether you want it in $D\flat$, in D or in $D\sharp$. This might become tricky if accidentals occur within the piece, which might have to be converted into flats, naturals, sharps or double sharps, depending on the new chosen signature. To avoid this trouble, *relative* accidentals have been implemented, the actual output of which depends on the pitch of this accidental and of the current signature.

1.3.3 Selecting special instrument scores

Another question is: *"can I write an orchestral score and extract the separate scores for individual instruments ?"* The answer is 95 % *yes*: in fact, you can define your own macros `\mynotes...\enotes`, `\myNotes...\enotes` with as many arguments as there are in the orchestral score (hope this is less or equal to 9, but TEXperts know how to work around) and change its definition depending on the selected instrument (or insert a test on the value of some selection register). But the limitation is that the numbering of instruments may change, so that `\signiii` may have to become `\signi` if instrument *iii* is alone. But, in turn, this is not a serious problem for average TEX wizard apprentices.

1.3.4 Variable staff and note sizes

Although the staff size is 20 pt as a standard, MusicTEX allors scores of 16 pt staff sizes. In addition, any instrument may have a special staff size (usually smaller than the overall staff size) and special commands `\smallnotesize` or `\tinynotesize` enable notes (and also beams or accidentals) to be of a smaller size, in order to quote optional notes or *cadenzas*.

1.4 How to get it

The whole *distribution* fits into a single 1.2Mbyte or 1.44Mbyte diskette. It can also be obtained through an *anonymous ftp* at `rsovax.lps.u-psud.fr` (193.55.39.100), after selecting the subdirectory `[.musictex]`. All sources (including fonts) are provided, either separately or "zipped" or as VMS "savesets".

1.5 Enhancements

1.5.1 Recent easy enhancements

Many enhancements have been asked for, and this is a proof that MusicTEX is considered as useful by many people. Some of these enhancements which seemed hard were in fact rather easy to implement, for example small notes to represent grace notes and cadenzas. But others may induce heavy problems, for example the need of having *nice* slurs and ties.

In the same way, we recently introduce Andreas EGLER's fonts to have nicer braces at the left of the score, for example for piano music.

Besides, we recently found a hard incompatibility of MusicTEX and a genuine product of the author's native country, namely `french.sty` by Bernard GAULLE which is the *standard* of the French TEX user group, namely the GUTenberg association. This was partly due to macro name collisions — easy to solve — and to the fact that `french.sty` sets a lot of characters as `\active` characters in order to make them handle correctly the French spacing before and after punctuation marks. This seems to have been solved in the 4.99 version of MusicTEX, with the restriction that French macros cannot be used — at least easily — within lyrics.

Besides, MusicTEX has been adapted — since version 5.00 — to score *gregorian chant* and *percussion music*.

1.5.2 The tie/slur problem

While typesetting notes and even beams is a rather simple problem because it is a *local typesetting*, ties and slurs are much more difficult to handle.

Of course there is small problem in case of a typesetter wanting a slur or a tie binding two consecutive notes, not separated by a bar. In practice this *very restricted* use of slurs or ties can easily be solved by putting some symbols extracted from the `slur16` or `slurn16/slurn20` fonts somewhere on the staffs using the general use `\zcharnote` macro.

But serious music typesetters or composers know that many ties are supposed to link notes which are on both sides of a bar, which is a likely place to insert line breakings, so that the coding of *ties* must have various versions and sizes to resist that possible line breaking. What has been said about ties is still more serious in the case of *phrasing slurs* which may extend over several bars, lines and sometimes pages. In this case, their shape is not only a question of producing a long curved symbol of nice looking shape, it also has to cope with *glue*. An then the worst is that music way of typing does not accept *ragged lines* but equal length lines, even for the last line of a music piece. Thus, long distance slurs and ties need to be cut into separate parts (beginning, continuing(s), endings) which TEX can only link using *horizontal line overlaps* or `\leaders` to insure slur continuity over this unavoidable glue.

Therefore and up to now, ties and slurs have been implemented in a way which may look rather ugly, but we think it is the only way of implementing *in one pass* ties and slurs which run *across glue*. The principle is to have tie/slur symbols with a rather long horizontal part.

Then, at each time a glue occurs and at each time a group of notes is coded while a slur or tie is pending, an `\hrule` is issued which overlaps the preceding tie/slur symbol so that the final output seems to contain a continuous line. Unfortunately, this is possible only in the glue expansion direction, namely in the horizontal direction.

Variable size initial and final curved slur symbols have recently been implemented; the user has to choose them according to his intention to have short or long range slur symbols.

1.5.3 Enhancement limitations

Many requested improvements have not been *yet* implemented for several reasons:

- The author's natural lazyness (!)
- More seriously: many of them would require using some more registers; unfortunately, TeX registers are not numerous (256 of each kind and the limit of `\dimen` registers is nearly reached) and we are afraid many requested new features would make TeX stupidly crash even when typesetting reasonable scores.
- We do not think it is wise to introduce in MusicTeX itself a great number of macros which would be poorly used by most users: the reason is that TeX memory is hardly limited and that unused macros may occupy some TeX storage which could make things crash because of `TeX capacity exceeded`...

1.6 Acknowledgements

The idea of implementing the present package is due to the previous work (MuTeX) of Andrea Steinbach and Angelika Schofer[1]. This work provided the basis of the Metafont codes and some line breaking procedures, which both are still used here... with 99% corrections and updates.

Besides, the original fonts of MusicTeX, named `musicn20`, `musicn16`, `musicn13` and `musicn11`, have been recently reviewed by Andreas Egler and the new release — used in the version 5.00 and above of MusicTeX — is called `musikn20`, `musikn16`, `musikn13` `musikn11` and `musicbra`.

[1] Steinbach A. & Schofer A., *Theses* (1987, 1988), Rheinische Friedrich-Wilhelms Universität, Bonn, Germany.

Chapter 2

Practical use

2.1 Heading statements

Before any reference to MusicTEX macros:

```
\input musicnft
\input musictex
```

which may be followed by `\input musicadd` in the case you have more than six instruments (voice is two instruments: one for the music, one for the text) or more than 6 simultaneous beams or ties or slurs.

After that, you may write a complete book of TEX provided that you do not use `&` as a tabulation character (its `\catcode` has been changed) inside the music score and that you do not overwrite MusicTEX's definitions. This means that no special macros have been designed to help you write titles, author names, comments, literature excerpts, etc., unless you use LATEX with the `musictex` style.

2.2 Before you begin to write notes

You should first specify whether you want to typeset music in size 20pt per staff or 16pt. This only optional, the default value being 20pt. If you want the 16pt size, then you have to say:

```
\musicsize=16
```

Then, the first compulsory declaration is:

`\def\nbinstruments{`n`}`

where n is the number of instruments, used by MusicTEX to performs loops building staffs, setting signatures, meters, etc. Therefore, it must be defined before any other statements. An instrument may consist of several staffs, e.g. the piano. The difference between one instrument of several staffs and several instruments is as follows:

- distinct instruments may have distinct *signatures*, distinct staffs of a unique instrument share the same signature.
- *stems* may be hung to *beams* belonging to differents staffs of the same instrument.
- *chords* may extend across several staffs of the same instrument.

- staffs of a unique *instrument* are tied together with a big brace at the beginning of each line.

If the number of staffs (in French "*portées*") is not equal to one, this number must be specified by:

`\nbportees`r`=`p`\relax`

where p is the number of staffs, and where r is the roman numeral of the instrument considered (e.g. `\nbporteesiii` for the 3rd instrument, starting from the bottom). This value may be zero, in which case the staff lines are omitted, and this instrument may be used to code the *lyrics* of a song, below the actual "instrument" representing the notes of the song.

Unless all your instruments only use the *violin* clef, you have to specifiy all the clefs used for all the instruments. This is done by coding:

`\cleftoks`r`={{`$s1$`}{`$s2$`}{`$s3$`}{`$s4$`}}%`

where r is the roman numeral of the instrument, $s1$ specifies the clef of the lower staff, $s2$ the clef of the second staff, etc. One must always give four values with the above syntax, otherwise... $s1 = 0$ means the *violin* clef (clef de sol in French), $s1 = 1$ through $s1 = 4$ mean the *alto* clef (clef d'ut in French) set on first (lower) through fourth (next to upper line of the staff), $s1 = 5$ means the *bass* clef at third (middle) line, and $s1 = 6$ means the usual *bass* clef (clef de fa in French) at the usual fourth line. As an example, a standard piano score should include:

`\cleftoksi={{6}{0}{0}{0}}%` or `\cleftoksi={6000}%`

If the signature is not void, one should code:

`\generalsignature{`s`}\relax`

where $s > 0$ indicates the number of *sharps* in the signature and $s < 0$ the number of *flats*[1].

If there a *meter* indication is to be posted, it should be specified using the macro

`\generalmeter{`m`}%`

where m is the description of the meter indication which should appear on each staff. If it is a *fraction* (e.g. 3/4) on should code

`\generalmeter{\meterfrac{3}{4}}%`

or, in a simpler way (if the numbers are less than 10):

`\generalmeter{\meterfrac 34}%`

Special denotations can be used, such as `\allabreve` to get ₵ and `\meterC` to get **C**.

However, not all music scores have the same meter in each staff. Especially, some staffs may have *ternary* meters while others have *binary*. This can be specified by using the `\generalmeter` macro to set the meter for most of the scores and overriding it by means of a more sophisticated command:

`\metertoksii={{\meterfrac{12}8}{\allabreve}{}{}}%`

which sets the meter to 12/8 for the first (lower) staff, and *alla breve* for the second staff of the instrument number 2 (*ii*). Note that there is room for 4 staffs and that void items must be specified, otherwise TEX weird errors occur.

[1]We have seen once a score in G-minor where the signature consisted of two flats (B and E) plus one sharp (F). This is not supported by MusicTEX.

2.2.1 Instrument names

If you want the *name of the instruments* (or the *name of the voices*) to be displayed in front of their respective staffs at the beginning, you may code:

`\def\instrument`*r*`{`*name of the instrument*`}%`

where r is the roman numeral of the instrument considered. In this case, you should also adjust the `\parindent` dimension so that the long name of an instrument does not spill too far into the left margin.

2.2.2 Polyphonic songs

Except staffs of a unique instrument tied together with a big brace, staffs normally begin on the left with a thin vertical rule. However, it is usual to tie all human voices together with a left heavy and right thin vertical rule. This can be specified (but only once per system) by specifying

`\def\lowersonginstrum{`m`}`
`\def\uppersonginstrum{`n`}`

where m and n are the intrument numbers of the first and last choral voices. An example of using this feature is given in `PRAETORI` and – more complicated – also in `ANGESCAM` and `ANGESCAO`.

2.3 Starting your masterpiece

2.3.1 Typing the first system

Just code

```
\debutmorceau
```

which will initiate (with indentation `\parindent`) the first set of staffs for all instruments you have previously defined. But that is not sufficient to begin writing notes and silences. In fact, you also must choose the spacing of the notes.

2.3.2 Easy selecting note spacing

The easiest way of getting a reasonable note spacing is done by saying

```
\normal
```

This defines an elementary spacing of `10pt`. If you say `\large`[2] the elementary spacing (`\elemskip`) is set to `12pt`. Once this is done, you can select $\sqrt{2}$ multiples of this value to select specific note spacing by initiating your note column with `\notes` (spacing `\elemskip`), `\Notes` (spacing `1.4\elemskip`), etc.

The is also a macro named `\etroit` which yields narrower[3] spacings but not increasing in the same way.

In practice, the choice of the macro `\notes`, `\Notes`, `\NOtes`, etc., to initiate of column of notes sets an internal dimension register, named `\noteskip` to the given multiple of `\elemskip`.

[2] To avoid problems with the LaTeX macro of the same name, this macro is only activated under LaTeX when `\begin{music}` is invoked.

[3] *Etroit* means *narrow* in French.

Thus, each *spacing note* (`\qu`, `\qh`, `\hl`, etc.) will be followed by a spacing of `\noteskip`. Then, the advantage of the definition of `\elemskip` is that, whenever it is changed, all subsequent `\noteskip`s will be updated proportionally so that a simple change of `\elemskip` can expand or shrink all consecutive note spacings as a whole.

If these values of `\elemskip` are not fit to your needs, you can also say

```
\normal\elemskip=15pt
```

or revert to the basic note introducing sequence:

`\vnotes` *q* `\elemskip` *note specifications* `\enotes`

where *q* is a positive integer or decimal number.

Of course, if you are sufficiently skilled with TEX, you can also compute `\elemskip` according to the number of bars you want in a line and the number of notes in each bar.

However, you may also prefer to ask MusicTEX to compute the size of `\elemskip` from the current page width (`\hsize`) after assuming a constant number of bars of a constant number of beats within each line[4]. In this case, you have better use the `\autolines` macro, which is described below (see: "Line and page breaking").

2.4 Note pitch specification

Note pitches are usually specified by letters ranging from `a` to `z` for those which are usually written under the G-clef (`a` corresponds to the *A* of nominal frequency 222.5 Hz; the *G* of the G-clef is denoted `g`). Lower pitch notes are specified using upper case letters ranging from `A` to `N` (the *F* of the F-clef is denoted `M`, and `F` is one octave below).

If necessary, a numeric symbol can be used to place a symbol independently of the active clef.

Besides, notes below `A` (i.e. the *A* of nominal frequency 55.625 Hz), namely the lowest octave of the modern pianos, can only be coded using the transposition features (see below: *transposition* and *octaviation*) or in absolute vertical position using numbers.

2.5 Writing notes

There are two major kinds of note macros:

1. those which terminate a note/chord stem and are followed by a horizontal spacing of value `\noteskip`,
2. those which initiate or extend a note/chord stem and do not cause horizontal spacing.

The first kind is used to type a melody, the second kind is used to type chords.

[4]This does not meet the requirements of contemporaneous music, but fits very well to baroque and romantic music.

2.5.1 Single (spacing) notes

`\wh` *p* : whole note at pitch *p*.

`\hu` *p* : half note at pitch *p* with stem up.

`\hl` *p* : half note at pitch *p* with stem down.

`\qu` *p* : quarter note at pitch *p* with stem up.

`\ql` *p* : quarter note at pitch *p* with stem down.

`\cu` *p* : eighth note[5] at pitch *p* with stem up.

`\cl` *p* : eighth note at pitch *p* with stem down.

`\ccu` *p* : sixteenth note at pitch *p* with stem up.

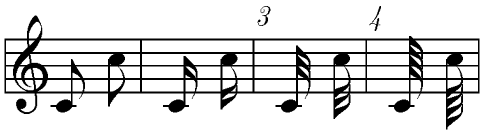
`\ccl` *p* : sixteenth note at pitch *p* with stem down.

`\cccu` *p* : 32-th note at pitch *p* with stem up.

`\cccl` *p* : 32-th note at pitch *p* with stem down.

`\ccccu` *p* : 64-th note at pitch *p* with stem up.

`\ccccl` *p* : 64-th note at pitch *p* with stem down.

As an example, the sequence:

was coded as:

```
\notes\cu c\temps\cl j\enotes\barre
\notes\ccu c\temps\ccl j\enotes\barre
\notes\cccu c\temps\cccl j\enotes\barre
\notes\ccccu c\temps\ccccl j\enotes
```

If these notes are preceded by *non-spacing* notes (i.e. macros `\zq` or `\zh`) their stem is extended up or down so as to join all notes into a single chord.

2.5.2 Non-spacing (chord) notes

`\zq` *p* : quarter (or shorter) note head at pitch *p* with no spacing after.

`\zh` *p* : half note head at pitch *p* with no spacing after.

It must be pointed out that the pitch *p* of these notes is memorized so that the stem of the further spacing note will join them into a chord. This stem top and bottom pitch is *reset* at each spacing note.

REMARK : Notes of duration longer than whole notes are always non-spacing. This saves one useless definition, since these notes are always longer than other simultaneous ones. If needed they can be followed by `\sk` *to force spacing.*

[5]The `\c` of this macro name is taken from the French word "croche" which is by the way one half of the english "crotchet"; `\cc...`, `\ccc...` are standing for "double croche", "triple croche", etc.

2.5.3 Shifted non-spacing (chord) heads

These symbols are used mainly in chords where *second* intervals are present. It is the responsibility of the typist to choose which heads should be shifted left or right.

`\rw` p : whole note head shifted right by one note width ($\approx$ 6pt), no spacing.
`\lw` p : whole note head shifted left by one note width ($\approx$ 6pt), no spacing.
`\rh` p : half note head shifted right by one note width ($\approx$ 6pt), no spacing.
`\lh` p : half note head shifted left by one note width ($\approx$ 6pt), no spacing.
`\rq` p : quarter note head shifted right by one note width ($\approx$ 6pt), no spacing.
`\lq` p : quarter note head shifted left by one note width ($\approx$ 6pt), no spacing.

Except that they are shifted left of right, these macros act like `\z...` macros for stem building.

2.5.4 Shifted notes

`\rqu` : acts like `\qu` but the note head is shifted one note width. This is used for *chords* with upper note on the right side of the stem.

`\rhu` : same as above for a half note.

2.5.5 Non-spacing single notes

`\zhu` : half note with stem up but no spacing. It acts like `\hu` for stem building.

`\zhl` : half note with stem down but no spacing. It acts like `\hl` for stem building.

`\zqu` : quarter note with stem up but no spacing. It acts like `\qu` for stem building.

`\zql` : quarter note with stem down but no spacing. It acts like `\ql` for stem building.

`\zcu` : eighth note with stem up but no spacing. It acts like `\cu` for stem building.

`\zcl` : eighth note with stem down but no spacing. It acts like `\cl` for stem building.

`\lhu`, `\lhl`, `\lqu`, `\lql` : same as above, but the whole of the note is shifted one note width on the left.

`\zw` p : whole note at pitch p with no spacing after.

`\zwq` p : arbitrary duration note () at pitch p with no spacing after.

`\zbv` p : breve note () at pitch p with no spacing after.

`\zsb` p : semi-breve note () at pitch p with no spacing after.

2.5.6 Pointed notes

One simple way of doing consists in putting `\pt` p to get a *dot* after the normal note head at pitch p. Thus a quarter note with a point can be coded `\pt h\qu h`.

A simpler way of doing consists in using compact macros, namely: `\qup`, `\qupp`, `\quppp`, `\zqp`, `\zhp`, `\zwp` (these three `\z...p` are useful in chords), `\hup`, `\whp`, `\qhp`, `\qhpp`, `\qlp`, `\qlpp`, etc.

You may also introduce pointed notes, especially in groups by coding a *period* before (not after) the letter representing the pitch: `\qu{.a.^b.c}` which is equivalent to:

```
\pt{a}\qu{a}\pt{b}\sh{b}\qu{b}\pt{c}\qu{c}
```

Finally, pointed nots can also be produced without spacing aftef, using `\zhup`, `\zhlp`, `\zqup`, `\zqlp`, `\zcup`, `\zclp`, and the same with two `p`'s for double-pointed notes.

2.6 Beams

Beams are not automatically handled, but they must be declared explicitely, *before* the first spacing note involving them is coded. Two kinds of macros are provided:

1. fixed slope beams have an arbitrary slope chosen by the user in the range -45% to +45% (by multiples of 5%);
2. semi-automatic beams have their slope computed knowing the number of `\noteskip` over which they are supposed to extend, and knowing the initial and final pitch of the notes they are supposed to link.

2.6.1 Fixed slope beams

`\ibu` *nps* : initiates an *upper beam* 3 horizontal line spacings above the pitch p ; m is its reference number, which must be in the range [0-5] ([0-9] if `musicadd` file has been `\input`); s is the slope of the beam.

s is an integer in the range [-9,9]. $s = 1$ means a slope of 5%, $s = 9$ means a slope of 45% (the maximum with the `beamn20` or `beamn16` fonts), $s = -3$ means a slope of -15%, etc. With usual spacings a slope of 2 or 3 is fit for ascending scales. A slope of 6 to 9 is fit for ascending arpeggios.

`\ibl` *nps* : initiates a *lower beam* 3 horizontal line spacings below the pitch p. Other parameters as above.

`\ibbu` *nps* : initiates a *double upper beam* (same parameter meaning).

`\ibbl` *nps* : initiates a *double lower beam* (same parameter meaning).

`\ibbbu` *nps* : initiates a *triple upper beam* (same parameter meaning).

`\ibbbl` *nps* : initiates a *triple lower beam* (same parameter meaning).

`\ibbbbu` *nps* : initiates a *quadruple upper beam* (same parameter meaning).

`\ibbbbl` *nps* : initiates a *quadruple lower beam* (same parameter meaning).

`\ibbbbbu` *nps* : initiates a *quintuple upper beam* (same parameter meaning).

`\ibbbbbl` *nps* : initiates a *quintuple lower beam* (same parameter meaning).

Beam termination is also not automatic. The termination of a given beam must be explicitely declared *before* coding the last spacing note connected to that beam.

`\tbu` n : terminates upper beam number n at current position.

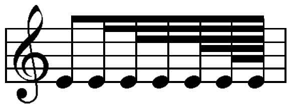
`\tbl` n : terminates lower beam number n at current position.

`\tbu` and `\tbl` terminate beams of any multiplicity. Therefore 32-th notes hanging on a triple beam are initiated by `\ibbbu` nps and terminated by `\tbu` n.

It is also possible to code beams whose multiplicity is not the same at the beginning. The multiplicity can be increased at any position. For instance, `\nbbu` n which sets the multiplicity of upper beam number n to 2 starting at the current position, `\nbbbu` n sets its multiplicity to 3, `\nbbbbu` n sets its multiplicity to 4, and `\nbbbbbu` n sets it to 5. `\nbbl` n ...`\nbbbbbl` n perform the same functions for lower beams.

Notes hanging or standing on beams are coded in the form `\qh`n p and `\qb`n p where n is the beam number and p the pitch of the note head. MusicTEX adjusts the length of the note stem to link the bottom of the chord to an upper beam (normally with `\qh`) and the top of the chord to a lower beam (normally with `\qb`).

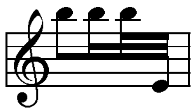
Note that the difference between upper and lower beams does not mainly consist in the beam being above or below the note heads; rather, it specifies whether the abscissa of the beginning and the end of this beam is aligned on the right (upper beam) or on the left (lower) beam. Thus, the sequence:



has been coded as

```
\notes\ibu0h0\qh0e\nbbu0\qh0e\nbbbu0\qh0e\nbbbu0\relax
\qh0e\nbbbbu0\qh0e\nbbbbbu0\qh0e\tbu0\qh0e\enotes
```

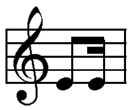
It is quite possible to terminate with `\tbu` a beam initiated with `\ibl`. This may give:



which has been coded as

```
\notes\ibl0p0\qb0p\nbbl0\qb0p\nbbbl0\qb0p\tbu0\qh0e\enotes
```

Partial termination of beams is also possible, by using `\tbbu` or `\tbbl` : these macros terminate the current beam except that of order 1 (eighths). `\tbbbu` or `\tbbbl` terminate the current beam except those of order 1 and 2, etc.

The macros `\tbbu` and `\tbbl` may also be invoked when only a single beam is active. Then, a second beam (upper or lower according the initiating procedure) is opened *one note width before the current position, and closed immediately*. Thus the following sequence

is coded:

```
\notes\ibu0e0\qh0e\tbbu0\tbu0\qh0e\enotes
```

The same behaviour occurs in the case of `\tbbbu`, `\tbbbl`, `\tbbbbu`, `\tbbbbl`, `\tbbbbbu` and `\tbbbbbl`.

The symmetrical pattern is also possible. For example:

has been coded as:

```
\Notes\ibbl0j0\rlap{\qsk\tbbl0}\qb0j\tbl0\qb0j\enotes
```

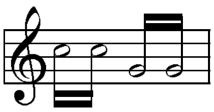
REMARK: these codings may seem complicated. In fact, it is the responsibility of the user to define macros which perform the most common sequences in his masterpiece. For example, one could define sets of four sixteenths by the macro:

```
\def\qqh#1#2#3#4#5{%
\ibbl0#2#1\qh #2\qh #3\tbl0\qh #4}
```

where the first argument is the slope and the other four arguments are the pitches of the four consecutive sixteenths wanted.

2.6.2 Repeated pattern beams

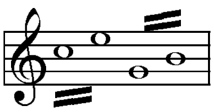
Note heads hanging on beams are not necessarily quarter (or higher order) note heads. It is possible to hang half note heads on beams using `\hh` and `\hb` macros, e.g.:



has been coded as:

```
\Notes\ibbl0j0\hb0j\tbl0\hb0j\enotes
\Notes\ibbu0g0\hh0g\tbu0\hh0g\enotes
```

It is also possible to write

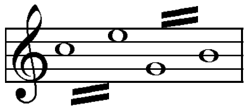


which was coded as:

```
\Notes\ibbl0j3\wh j\tbl0\wh l\enotes
\Notes\ibbu0g3\wh g\tbu0\wh i\enotes
```

However, a better look could be obtained in a more sophisticated way[6]:

[6] You are suggested to make your own macro if you have to type many of these.



which was simply coded as:

```
\Notes\zw j\qsk\ibbl0j3\sk\tbl0\wh l\enotes
\Notes\ibbu0g3\wh g\tbu0\qsk\wh i\enotes
```

2.6.3 Semi-automatic beams

In order to avoid tedious checks to adjust the slope (and even the starting pitch) of beams in music with a lot of steep beams, a set of automatically slope computing has recently been implemented (file `musicvbm.tex`). If you say `\Ibu2gj3` MusicTEX will understand that you want to build an upper beam (beam number 2) horizontally extending 3`\noteskip`, the first note of which is a `g` and the last note is `aj`. Knowing these parameters it will choose the highest slope number which corresponds to a slope not more than (`j` − `g`)/(3`\noteskip`). Moreover, if there is no sufficiently steep beam slope available, then it will raise the starting point.

Ten such macros are available: `\Ibu`, `\Ibbu`, `\Ibbbu`, `\Ibbbbu`, `\Ibbbbbu`, `\Ibl`, `\Ibbl`, `\Ibbbl`, `\Ibbbbl` and `\Ibbbbbl`. Examples of their use is given in `marcello.tex`.

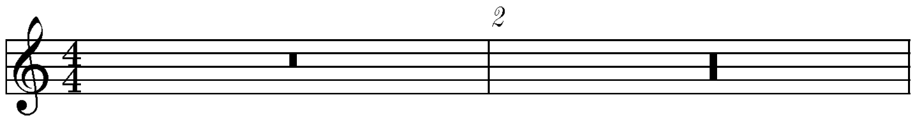
2.7 Rests

Except that difference that they have no specific pitch, rests are coded in a very simple way.

2.7.1 ordinary rests

Full bar rests (also called “pauses”) are coded as `\pause`; smaller rests are `\hpause` (of duration equal to a *half note*), `\soupir` (duration equal to a *quarter note*), `\dsoupir` (duration equal to an *eighth note*, also `\ds`), `\quartsoupir` (duration equal to an *sixteenth note*, also `\qs`), `\huitsoupir` (duration equal to an *thirtysecond note*, also `\hs`), `\seizsoupir` (duration equal to an *sixty fourth note*, also `\qqs`).

Long rests (lasting several bars) can be coded as `\PAuse` and `\PAUSe`, which respectively yield:



2.7.2 Lifted rests

All the previous rests are *hboxes*; this means that they can be vertically offset to meet polyphonic music requirements using the standard TEX command `\raise`.

In addition, two symbols have been provided to put a *full rest* or a *half rest* aboce or below the staff. Then the ordinary `\pause` or `\hpause` cannot be used since there is a need for small horizontal line to distinguish between the full and the half rest. They are :

- `\expause` *p* (spacing) for at pitch position *p*,

- `\exhpause` *p* (spacing) for 𝄼,
- `\zexpause` *p* (non spacing) for 𝄻,
- `\zexhpause` *p* (non spacing) for 𝄼.

2.8 Phantom notes

It may be interesting, when coding a sequence of notes within a unique pair `\notes...\enotes`, to skip one note place in order – for example – to set the third note of one staff at the same abscissa as that of the second note of another staff. This can be done by inserting `\sk` which causes a spacing of one `\noteskip`[7].

If you just want to shift a note or a symbol by one note head width, you may write `\qsk`.

Note that these two latter macros must be used inside a pair `\notes...\enotes`. If you want to make a spacing of one note head width outside, write `\nspace`. Conversely, a more general spacing is allowed within the `\notes...\enotes` pairs, namely:

`\off` *d*

where *d* is any TeX valid dimension, for example `\noteskip` or `5\Interligne`. In fact, if you look to the MusicTeX source, you will find that `\off` is the basic control sequence used to define `\sk`, `\qsk`, etc.

2.9 Collective coding: sequences of notes

As seen in the MOZART example, it is not necessary to write a macro sequence `\notes...\enotes` for each column. If, on all staffs of all instruments, spacings are equal or multiple of a unique value, the notes may be concatenated in each staff: each note in each staff makes the current position horizontally advance by the elementary spacing specified by the choice of `\notes`, `\Notes`, `\NOtes`, etc.

The major interest of this feature resides in that fact that the note macros are able to write several items; for instance `\qu{cdefghij}` writes the *C-major* scale in quarters with stem up. In the same way `\cl{abcdef^gh}` writes the *A-minor* scale in eighths. Not all note generating macros can be used to perform collective coding, but most of them can. They are:

- all the spacing notes: `\wh`, `\hu`, `\hl`... `\ccccl` and the beam hooked notes, i.e. `\qh`*n* and `\qb`*n*.
- all the chord notes with names of the form `\z...`.

Conversely, shifted notes are not supposed to be used in collective coding, mainly because they are used in very special cases which do not deserve wasting memory space to make them collective.

If necessary a void space can be inserted in a collective coding by using `*`.

[7] Do not use `\kern` nor `\hskip` : in fact `\sk` not only causes a space but also records that space for correct handling of beams.

2.10 Accidentals

Accidentals can be introduced in two ways.

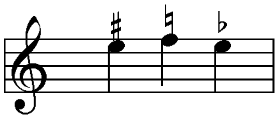
The first way, the *manual* way of coding them, consists for example in coding `\fl a` to put a *flat* at the pitch *a*, supposedly before the further note of that pitch. There is no control upon the fact that a note will be put at this position and at this pitch. Naturals, sharps, double flats and double sharps are coded `\na` *p*, `\sh` *p*, `\dfl` *p* and `\dsh` *p* respectively.

Alternate procedures `\lfl`, `\lna`, `\lsh`, `\ldfl` and `\ldsh` place the same accidentals, but their abscissa is shifted one note head width on the left. The purpose of this is to avoid collision of accidentals in a chord with narrow intervals.

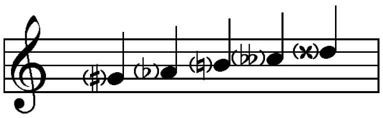
The second way of coding accidentals consists in putting the symbol `^` (sharp), the symbol `_` (flat), the symbol `=` (natural), the symbol `>` (double sharp), or the symbol `<` (double flat) within the coding of the note, e.g.: `\qh{^g}` yields a $G\sharp$. This may very well be combined with collective coding, e.g.: `\qu{ac^d}`.

Two sizes are available for accidentals. They revert to the small version when notes are supposed to be too close to each other. These two sizes can be forces by coding `\bigfl`, `\bigsh`, etc., or `\smallfl`, `\smallsh`, etc. If one does not want to have any small accidentals, then one can declare `\bigaccid` (conversely `\smallaccid` or `\varaccid` – the latter restoring variable sizes).

Small accidentals can also be put *above* the note heads. This is done using `\uppersh` *p*, `\upperna` *p* or `\upperfl` *p*:



It also possible to introduce *cautionary accidentals* on a score, i.e. small size accidentals between parentheses. This done by preceding the name of the accidental keyword by a `c`, e.g. by coding `\cfl`*p* to get a cautionary flat. Available cautionary accidentals are `\csh`, `\cfl`, `\cna`, `\cdfl` and `\cdsh`, which give:



Since cautionary accidentals are wider that normal ones, it might be useful to insert some spacing before them, either using `\temps` or a `\sk` within a `\notes...\enotes` group.

2.11 Transposition and octaviation

An important feature is the existence of a special register `\transpose` the normal value of which is 0. If you say

```
\transpose=3
```

all subsequent pitches specified by upper or lower case letters will be transposed 3 positions. If you set `\transpose` to 7 you may write your music one octave below its final pitch. Thus, you can define *octaviation* macros like

```
\def\soqu#1{\zq{#1}{\transpose=7\relax \qu{#1}}}
```

to build quarter note octaves in a single call. Note that the octaviated note is coded within braces so that the transposition is only local.

Octaviation can also be performed in is another way, namely unsing special codes to transpose by multiples of 7 intervals. For example `\qu{'ab}` is equivalent to `\qu{hi}` and ``\qu{`kl}`` is equivalent to `\qu{de}`. It should be emphasized here that the `'` (*acute accent*) and the `` ` `` (*grave accent*) have cumulative effects, so that `\qu{''A'A}` is equivalent to `\qu{ah}` and that the `\transpose` parameter is only reset to its initial value (not necessarily zero) when changing staff or instrument (i.e. `|` or `&`) or at `\enotes`. Since this may be confusing, it is useful to use the `!` prefix to reset the `\transpose` register explicitely to the value it had when entering `\notes`[8]. Thus `\qu{!a'a}` always gives the note `a` and its upper octave `h` *shifted by the value of* `\transpose` *at the beginning of the current* `\notes...\enotes` group (or `\Notes...\enotes`, etc.) whatever the number of previous grave and acute accents occurring inbetween.

The above processes indeed change the vertical position of the note heads and associated symbols (note stems, accents and beams) but they do not take care of the necessary changes of accidentals when transposing, i.e. the fact that an $F\sharp$ occurring with a zero signature should become a $B\natural$ when transposing from the tonality of C major to F major where the normal B is the $B\flat$. Since the intent of the composer is not obvious – he may want to shift a group of notes within the same tonality or conversely to transpose it in another tonality – this is not done automatically. Thus the `\sh`, `\fl`, `\na`, `\dsh` and `\dfl` symbols *are not affected* by a change of the `\transpose` register.

But the composer/typesetter may ask MusicTEX to do that work. In this case, he should code `\Sh`, `\Fl`, `\Na`, `\dSh` and `\dFl` (or `\bigSh`, `\bigFl`, `\bigNa`, `\bigdSh` and `\bigdFl` or `\smallSh`, `\smallFl`, `\smallNa`, `\smalldSh`, `\smalldFl`, `\cSh`, `\cFl`, `\cNa`, `\cdSh` and `\cdFl`) instead of the usual lower case accidental symbols. The symbol `\Sh` (resp. `\bigSh` and `\smallSh`) means that the corresponding pitch has to be raised by *one half pitch* with respect to its normal value *according to the current signature*. Thus `\Sh b` means a $B\sharp$ if the signature is zero or positive, and a $B\natural$ if it is negative. The same logic applies for all accidentals having an upper case forelast letter.

Obviously, the computation is done after taking account of the value of the `\transpose` register.

The compact codes `^`, `_`, `=` are normally not affected by transposition and signatures, but their behaviour can be changed by saying `\relativeaccidentals` and reset by `\absoluteaccidentals` (the default situation).

Although *relative accidental coding* is an easy and safe way of coding *transposable* scores, care should be exercised in getting rid of the habit of saying `\na b` to rise the pitch of a B when the tonality is F major (i.e. with `\sign`n`=-1` or `\generalsignature{-1}`). An example of sophisticated transposition is given in the score `souvenir.tex` (which `\inputs souvenix.tex`). It should be noted that *relative accidental coding* is compatible with *cautionary accidental* coding: in this case, one should code `\cNa`, `\cSh`, `\cFl`, `\cdSh`, `\cdFl` respectively.

Besides, the typical piano octave transposition *8 - - - - -* can be obtained by coding:

`\octfin` p n

[8]This value is saved in another register named `\normaltranspose`.

which puts the 8 and dotted line symbols at the pitch p (usually `p` to `r`). The length is n`\noteskip`. This obviously fit for short octaviation denotations. To transpose a whole line, use `\octline` p. Since `\octfin` terminates with a small hook down, to indicate clearly where octaviation stops, you may also like to use `\octsup` which behaves like `\octfin` without the final hook. All this supposes you have an idea of the actual line breaking of your score in that section, and this is admittedly difficult to handle when octave transposition is supposed to long a large number of lines. In that latter case, you can use the `\def\everystaff{...}` to insert whatever code you like at each new line of score. This has been used to set octaviation in the score of the *Toccata in F* by Charles-Marie WIDOR[9]: at each bar (or virtual bar, namely `\zbarre`), the `\everystaff` procedure is updated to produce the convenient code, in case of the line breaking happening before the next definition.

2.12 Ties and slurs

They have been implemented in a way which may look rather ugly, but we think it is the only way of implementing *in one pass* ties and slurs which run *across glue*.

Slurs and ties must be initiated within the pair `\notes...\notes` before the spacing note is coded. They must be terminated also before the last note is coded.

`\itenu` np

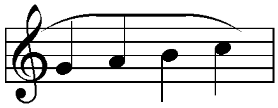
(*ten* stands for the Italian word *tenuto*) initiates an upper tie (convex) at pitch p. Just like beams, ties have a reference number n, from 0 to 6 (or 9 if `musicadd` is included). `\itenl` np initiates a lower tie (concave).

The tie of reference number n is terminated by `\tten` n.

Slurs are initiated with `\ilegu` np and `\ilegl` np, where *leg* stands for the Italian word *legato*, and they are terminated with `\tleg` n. Except that slurs start before the note position and stop after, ties and slurs work the same way with the same syntax. They also share the same registers so that slurs nesting tied notes should have numbers n distinct from each other.

2.12.1 Standard and nicer slurs

Since version 4.7, beginnings and ends of slurs are produced using a special font (`slurn16` or `slurn20`) rather than using TEX bracing tokens from `cmex10`. Thus the bending of slurs is now nicer and, moreover, several slur sizes are available. For example:



has been coded using `\Ilegu` rather than `\ilegu` (resp. using `\Ilegl` rather than `\ilegl` if a lower slur is wanted), namely:

```
\NOtes\Ilegu0l\qu g\enotes
\NOtes\qu h\enotes
\NOtes\ql i\enotes
```

[9]A French organist (1844-1937) and composer who was in charge of the organ of St-Sulpice in Paris, from 1864 to 1934.

```
\NOtes\tleg0\ql j\enotes
```

It should to be noted that, as a standard, *ties* still use the TEX bracing symbols; this is mainly done to avoid visibility problems with ties occurring in the middle of staffs, i.e. in a situation where tie lines could overlap and be confused with staff lines. But the MusicTEX user can request using special slur fonts by coding `\Itenu` or `\Itenl` instead of `\itenu` or `\itenl`. Even more, big symbol ties are available using `\ITenu` or `\ITenl`.

As a standard, only two slur and tie symbol sizes are referred by MusicTEX macros, but a quick look into `musicnft.tex` show that these definitions explicitely refer to a character number of the slur font:

```
\def\ITenu#1{\selecttenue{#1}\I@tenu{70}}%
\def\Ilegu#1#2{\selecttenue{#1}{\advance\locx@skip by -1.5\qn@width
\I@tenu{70}{#2}}}%
\def\Itenu#1{\selecttenue{#1}\I@tenu{66}}%
\def\ilegu#1#2{\selecttenue{#1}{\advance\locx@skip by -1.5\qn@width
\I@tenu{66}{#2}}}%
```

Then the MusicTEX user can, for his own special needs, define new macros using other character codes in the range 64–79, knowing that each character of this range is one point longer than the previous one.

2.12.2 Problem with large slur ends

Due to spacing problems, it may happen that the end of an `\Ilegu` or `\Ilegl` makses some ugly overlap between the currrent slur horizontal rule and the ending symbol itself. This is due to the ending symbol being so long that it overlaps the end of the horizontal rule. The easiest way of removing that slight trouble consists in replacing — for example — `\tleg0` with the more sophisticated commands :

```
\rlap{\qsk\tleg0}
```

which delays the position of the closing slur symbol by a space equal to the note head width.

2.12.3 Weird slurs and ties

In some special cases, usually in romantic music, one may want to invert the sense of a slur, linking for example a keyboard phrase beginning at the left hand and ending at the right hand. Thus, the slur of number n (a numeric vamue, not a roman numeral) can be inverted by

`\Invertslur` n

Besides, for some good reason, one may want to raise a slur at the next line. This is done by

`\def\atnextline{\liftslur` n`{` *dimension* `}}`

which will lift the slur number n by the given amount at the next line change. Of course, `\liftslur` could be called in the middle of a line, but the result would be rather ugly.

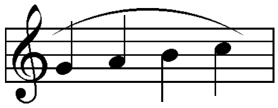
These two features are used at bars number 131–132 of examples `PACIFIQB` and `PACIFIQN`.

2.12.4 Refined slurs for clever typesetters

In addition to the previous extension and improvement of existing features, long nice horizontal slurs are also available, but they can be used only with two restrictions:

- They are not managed when *glue* is inserted, especially by `\temps` and `\barre`;
- They cannot be cut at line breaking.

Thus, their use is recommended only for nice slurs extending a few bars and not supposed to occur across lines. With this restrictions,



has been obtained by coding:

```
\NOtes\zcharnote l{\huslur{3.5\noteskip}}\qu g\enotes
\NOtes\qu h\enotes
\NOtes\ql i\enotes
\NOtes\ql j\enotes
```

2.13 Bars

2.13.1 Bars and spacing

Ordinary *bars* a coded using the macro `\barre` (this is a French word[10]). This macro provides an optional (discretionary) line or page break[11]. It also provides some *glue* in order to expand the text over an evenly filled line.

However, since the number of bars in a score line is generally small, it may be convenient to allow *glue* not only on each sides of *bars*. This can be done using the macro `\temps` (the French word for *beat*). This macro has two effects:

1. it inserts some *glue* but prevents line breaking[12],
2. if some ties or slurs are pending it expands them across the glue by writing an `\hrule` which overlaps the unfinished tie and makes it look prolongated.

Whatever the care you exercize in adjusting the size of the `\elemskip`, you are still likely to find some broken ties (which indicate excessive glue disassembling the gliding tie `\hrule`s) or some unexpected (and unwanted) line breaks or some Over[aw]full `\hbox`es. A useful means of estimating the remaining space to be filled with glue consists in declaring `\raggedlinestrue` : after that, an `\hfil` will be inserted by MusicTEX before each computed (when using `\autolines`) or forced line break. Thus, all the musical text will be packed on the left of the line and you will clearly see the amount of remaining space. Then, it will be up to you do decide changing some spacing parameters.

[10]Whose advantage is that it differs from `\bar` which is already defined in TEX.
[11]Unless it is triggered according to bar counting when `\autolines` has been invoked.
[12]This could unpleasantly occur if you insert a space...

Important: *do not use* `\temps` *when beams are pending, otherwise their spatial synchronization would fail. In other words, ties and slurs can jump over glue (because horizontal rules may overlap and thus have some elasticity) but beams (as well as any oblique lines) cannot.*

2.13.2 Bar numbering

Unless otherwise specified, bars are numbered. This is a good means of finding errors provided that the MusicTEX user has put comments in his source text recording the (expected) bar number. However, this can look unpleasant for final outputs, since the habit is to number bars only each other five or ten bars. This is not a serious problem since the frequency of bar numbering is defined as:

```
\def\freqbarno{1}
```

If you replace the `1` by `5`, bar numbering will occur each other five bars. You can also inhibit any bar number printing by telling:

```
\def\wbarno{}
```

or, in a more clever way:

```
\def\freqbarno{9999}.
```

The bar counter is also accessible, its name is `\barno`. This has nothing to do the the bar counting invoked by `\autolines`, so you can change it without any dramatic consequence.

2.13.3 Full and instrument divided bars

Normally, bars (as well as double bars, final bars and repeat bars) are drawn as a continuous line, starting for the bottom of the lower staff of the lower instrument, and ending at the top of the upper staff of the upper instrument. However, one may want to have discontinuous bars, that is, one continuous bar for all the staffs of a unique instrument. This is done by issuing the command `\sepbarrules`. An example of this is given in the `ANGESCAO` (or `ANGESCAM`) example; it has also been used in the example of section 2.19.2.

The initial situation can be forced or restored by `\stdbarrules`.

2.14 Line and page breaking

Bars provide a line breaking mechanism which is supposed to enable TEX to break a full score into lines and pages, with an optimal distribution of the text into lines and pages. Unfortunately, this does not work correctly for scores of more that approximately one page. The reason is that TEX must compile the entire contents of a paragraph, before it tries to break it into lines and pages. Therefore, one cannot rely on TEX to make that work which automatically results in the diagnostic `TeX capacity exceeded, memory....`

To circumvent this dramatically restrictive capacity[13] unless you use some BigTEX[14] another mechanism must be invoked to break lines.

The first one is the manual one: you replace some of the `\barre` macro calls by either `\alaligne` (equivalent to `\break` within text: in fact it contains a `\break` plus some (many)

[13]TEX has been designed to type text, not music.
[14]Whose drawback is that it is very slow on ordinary PCs.

other things). In the same way, you can code `\alapage` to force an `\eject` with proper reinitialization of staffs, clefs and signatures.

The second one is fit for scores with bars of regular length: after `\debutmorceau`, you code the following macro:

`\autolines` *tml*

where t is the number of *elementary spacings* (the length of `\notes...\enotes`) in an average bar, m is the number of bars you wish in a line and l is the number of lines you wish in a page[15].

This sets some parameters, namely `\maxbarsinline` and `\maxlinesinpage` which are simply used to count the bars, optionally perform `\alaligne` or `\alapage` instead of the normal `\barre`. You may freely alter the values of these parameters, once they have been established by `\autolines`. Moreover, you can still force line breaking of page ejection using `\alaligne` or `\alapage` without problem since these macros actually reset the bar counters appropriately.

On the other hand, you may want to forbid line breaking at a bar, then you replace `\barre` by `\xbarre`.

Conversely, you may want to break a line *not at a bar*[16]. This is allowed by `\zbarre` (optional line break) or forced by `\zalaligne` or `\zalapage`.

The final heavy double bar of a piece is provided by `\finmorceau`. If you just want to terminate the text with a simple bar, you say `\suspmorceau`. If you want to terminate it without a bar, you code `\zsuspmorceau`. Once you have stopped the score by any of these means, you can restart it using `\reprmorceau`.

If you want the next vertical bar to be a double bar, you have to declare `\setdoublebar` before the `\barre` (or the `\suspmorceau` or `\alaligne` or `\alapage`) to be marked with a double thin bar.

2.15 Changing score attributes

As seen before, you can change the signature of the whole set of instruments by `\generalsignature` n where $n > 0$ means a number of sharps, $n < 0$ means a number of flats. Or, you may prefer to change the signature of only one or two instruments by the statement:

`\sign`*r*`=`*s*

where r is the roman numeral of the instrument considered, and s its specific signature. Since you may change simultaneously (with respect to the score) but consecutively (with respect to your code) the signatures of several instruments, this change takes place only when you say `\changesignature` (within a bar) or `\changecontext` (after a single vertical rule) or `\Changecontext` (after a double vertical rule). In the same way, you may want to change the active clefs. This is done by

`\cleftoks`*r*`={{`*s1*`}{`*s2*`}{`*s3*`}{`*s4*`}}%`

where r is the roman numeral of the instrument, $s1$ specifies the clef of the lower staff, $s2$ the clef of the second staff, etc. One must always give four values with the above syntax, otherwise...

[15] After having coded nearly one hundred of pages of music, I strongly recommend the use of `\autolines` except when inserting short excerpts of less than one line, such as in musicographic books.

[16] For example, you may prefer to turn the page at a place where the pianist has one hand free.

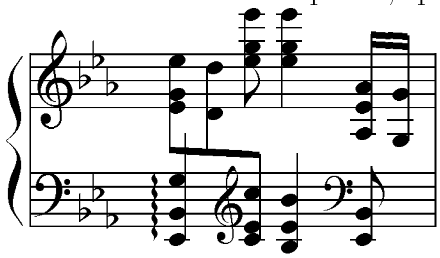
$s1 = 6$ means the *bass* clef (clef de fa in French), $s1 = 0$ means the *violin* clef (clef de sol in French), $s1 = 1$ through $s1 = 5$ mean the *alto* clef (clef d'ut in French) set on first (lower) through fifth (upper line of the staff). As seen above in the case of signatures, several clefs may be changed at the same time; thus all the clef changes become operational only when the macro `\changeclefs` is coded. Normal usage consists in issuing this command before the bar, not after (this helps the music player when the change happens across a line break).

Caution should be exercized when changing clefs when *beams* are pending: in fact the `\changeclefs` perturbate the computation of beams and this command should be invoked only

- out of `\notes`, `\Notes...` — `\enotes` pairs;
- only when no beam is pending[17].

If a clef change has to be typeset when one or several beams are pending, the operation should be done in a more manual way:

- say `\saveclefs` instead of `\changeclefs` : This actually records the `\cleftoks`*r* statement and no further `\changeclefs` will repeat the clef change, but the new clef will not pe typeset;
- say `\charnote0{\smallclefdesol}` (resp. `\charnote0{\smallclefdefa}` or `\charnote0{\smallclefdut}`) where you like it and at the right staff (i.e. within `\notes`, `\Notes...` — `\enotes` pairs) to type set the wanted clef symbol.

As an example, an excerpt of Brahms's Intermezzo op. 117,1 provided by Miguel Filgueiras:

which was coded as:

```
\begin{music}
\def\nbinstruments{1}\relax
\def\freqbarno{9999}% no bar numbers
\nbporteesi=2\relax
\cleftoksi={6000}\generalsignature{-3}% 3 flats
\debutextrait
\normal
\off{1em}\temps\Notes\larpeggio{E}5\zq{EI}\qu{N}%
\charnote0{\smallclefdesol}\relax
\nextstaff\qsk\ibl0e{-1}\zq{eg}\qb0l\zq{d}\qb0k\enotes
\cleftoksi={0000}\saveclefs\Notes\tbu0\zq{ce}\qh0j\relax
\nextstaff\zq{sn}\cl{l}\enotes
\Notes\zq{be}\qu{i}\nextstaff\zq{sn}\ql{l}\enotes
\cleftoksi={6000}\changeclefs\Notes\zq{E}\cu{I}\relax
```

[17] No problem with slurs and ties.

```
\nextstaff\ibbu1h{-1}\zq{ae}\qh1h\tbu1\zq{N}\qh1g\enotes
\finextrait
\end{music}
```

Meter changes are implemented the same way:

\metertoks$r$={{$m1$}{$m2$}{$m3$}{$m4$}}%

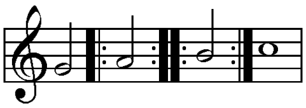
where r is the roman numeral of the instrument, $m1$ specifies the meter of the lower staff, $m2$ the meter of the second staff, etc. One must always give four values with the above syntax, otherwise... Since meter changes are meaningful only across bars, they are actually taken in account with \changecontext or \Changecontext or \alaligne or \alapage.

2.16 Repeats

To insert a *repeat bar* you can use several sets of procedures.

2.16.1 Elementary and unsafe solutions

The simplest consists in using the commands \leftrepeatsymbol, \rightrepeatsymbol and \leftrightrepeatsymbol – coded outside the \notes...\enotes pairs – which will simply insert these colon adorned double bars at the requested place. For example:



has been coded as:

```
\notes\hu g\enotes
\leftrepeatsymbol
\notes\hu h\enotes
\leftrightrepeatsymbol
\notes\hu i\enotes
\rightrepeatsymbol
\Notes\wh j\enotes
```

However, the previous way of coding does not provide for line breaking at repeat bars, nor does it advance the bar numbering. In fact, this way of coding if only fit for repeats occurring in the middle of a bar.

A second way of coding consists in saying \setleftrepeat, \setrightrepeat or \setleftrightrepeat before a bar (\barre), \suspmorceau or \changecontext). In this case, the next single[18] vertical bar will be replaced with the selected repeat bar. This meets the traditional music typesetting conventions in the only case of the *right repeat* but, unfortunately, left and left/right repeats use to behave in a different manner when in the middle of a line and at a line break.

[18]\setrightrepeat does not change the effect of \finmorceau; in that case use \finrightrepeat.

2.16.2 Safe and correct coding

The third coding, namely the correct coding – i.e. transforming in the correct manner when occurring at a line break – is obtained by substituting the `\barre` command with `\leftrepeat`, `\rightrepeat` and `\leftrightrepeat`.

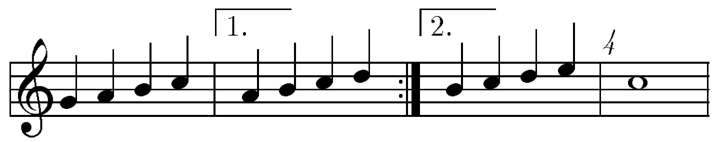
Now, if you want to force a new line at a repeat, you should code respectively:

```
\setrightrepeat\alaligne
\setrightrepeat\suspmorceau
\alaligne\leftrepeatsymbol
\reprmorceau\leftrepeatsymbol
\debutmorceau\leftrepeatsymbol
```

or the combination of two of these in the case of a left/right repeat.

2.16.3 Specific first and second pass scoring

A frequent situation consists in a long repeated score, but the last few bars are different at first pass and at second pass. This can be specified by saying `\setprimavolta` or `\setsecondavolta`[19] before the next `\barre` or *repeat* (usually `\rightrepeatsymbol`). For example :



has been coded as:

```
\def\nbinstruments{1}
\generalsignature{0}\savesignature
\nbporteesi=1\cleftoksi={{0}{0}{0}{0}}\relax
\debutextrait\normal
\Notes\qu {ghij}\enotes
\setprimavolta\barre
\Notes\qu {hijk}\enotes
\setsecondavolta\setrightrepeat\barre
\Notes\qu {ijkl}\enotes
\barre
\NOTes\wh j\enotes
\finextrait
```

The "prima volta" and "seconda volta" symbols are normally set at `2\Interligne` above the upper line of the staff, and one centimeter wide. If the music typesetter does not like that, he can issue for example:

```
\Setprimavolta{4\Interligne}{0.78in}
```

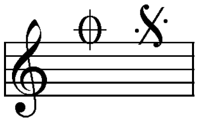
where the first argument is the vertical offset, and the second one is the length of the upper horizontal line. A symmetrical feature is `\Setsecondavolta` with also two dimension arguments.

Note that these special settings are only valid for one usage; after that, they are reset to the standard values.

[19] From the Italian *prima volta* meaning "first time", and *seconda volta* meaning "second time".

2.16.4 Large scope repeats

Large scope repeats have also been provided special symbols, namely using `\coda` and `\segno` :



which have been coded:

```
\NOtes\coda l\enotes
\NOtes\segno l\enotes
```

2.17 Miscellaneous

2.17.1 Putting anything anywhere

Special macros are provided to help the composer to set any TEX text on the staffs. The macro

`\charnote` *p*`{` text `}`

sets the given text with its base line at pitch *p* of the current staff (this means it must be coded inside `\notes...\enotes`). Whatever the length of the text, the spacing is `\noteskip`. If you do not want it to cause spacing, you code `\zcharnote`. If you want the possible spilling text to expand on the left rather than on the right, then you can use `\lcharnote`.

To place some text at the mid-position between the two staffs of a keyboard instrument, you may code:

`\midtwotext{` text `}  %` (spacing)

`\zmidtwotext{` text `}  %` (non spacing)

being however careful, a) to put it inside `\notes...\enotes`, b) to code it in the text of the lower staff.

A text to be put above the current staff is introduced by `\uptext{...}`. This may however cause some collision with bar numbering or notes above the staff; it is then wise to use `\Uptext{...}` which puts the text two pitches higher (recommended to post the tempo).

The macro `\zcharnote` is fit for coding special notations like accents above or below the notes. It behaves like `\charnote` but causes no spacing.

2.17.2 Metronomic indications

Metronomic indication deserves a special macro. The mention:

 = 60

is coded by `\metron{\hup}{60}` (normally embedded in `\Uptext` which is in turn embedded within `\notes...\enotes`).

2.17.3 Usual ornaments

Arpeggios (i.e.) can be coded with the macro

`\arpeggio` *pm*

where p is the pitch of the base of the arpeggio symbol and m is its multiplicity (one period is equal to one space between staff lines, i.e. 5 points). This macro causes a space of one note head width. If should be issued before the concerned chords. Its variant `\larpeggio` sets the arpeggio symbol slightly more on the left, in order to avoid collision with accidentals in front of the chords.

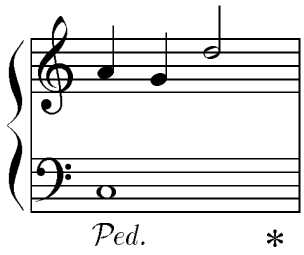
Note the `\arpeggio` and `\larpeggio` make an extra spacing of width one notehead (i.e; `\qsk`) so that it is wise to insert a `\qsk` in the other staffs before notes which are syncrhonous to the arpeggiated note.

Trills can be coded in several ways. `\trille` l (where l is a TEX dimension) yields while `\Trille` l (where l is a TEX dimension) yields *tr* . To put these patterns at a given pitch, one may use `\xtrille` *pl* or `\xTrille` *pl*. On the other hand `\ntrille` *pn* is equivalent to `\xtrille` *p*`{`*n*`\noteskip}` and `\nTrille` *pn* is equivalent to `\xTrille` *p*`{`*n*`\noteskip}`

Other *ornaments* are available:

- `\mordant` p for ,
- `\pince` p for ,
- `\Pince` p for ,
- `\Lpince` p for (thanks to A. EGLER),
- `\Pincesw` p for ,
- `\Pincene` p for ,
- `\Pincenw` p for ,
- `\turn` p for ,
- `\backturn` p for ,
- `\upz` p (upper *pizzicato*) to put a dot above a note head at pitch p,
- `\lpz` p (lower *pizzicato*) to put a dot below a note head at pitch p,
- `\usf` p (upper *sforzando*) to put a > accent above a note head at pitch p,
- `\lsf` p (lower *pizzicato*) to put a > accent below a note head at pitch p,
- `\ust` p (upper *staccato* or *portato*) to put a hyphen above a note head at pitch p,
- `\lst` p (lower *staccato* or *portato*) to put a hyphen below a note head at pitch p,
- `\uppz` p (upper strong *pizzicato*) to put an apostrophe above a note head at pitch p,
- `\lppz` p (lower strong *pizzicato*) to put a reversed apostrophe below a note head at pitch p.
- `\pointdorgue` p puts a *fermata* (in French "point d'orgue") at pitch p. No spacing occurs.

- `\pointdurgue` *p* puts a reverse *fermata* at the same place.
- `\PED` to insert the piano pedal command below the staff; pedal release is specified by `\DEP`; thus the following example



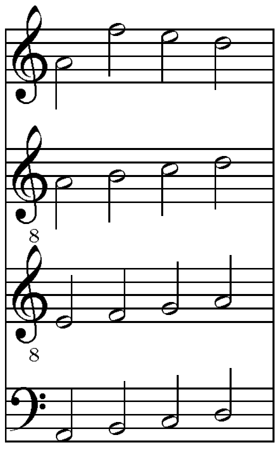
was coded as:

```
\temps\NOtes\qsk\PED\wh J|\qsk\qu h\enotes
\temps\NOtes|\qu g\enotes
\temps\NOtes|\hu k\enotes
\temps\Notes\DEP\enotes
```

2.17.4 New line synchronization of coding

The procedure named `\everystaff` is executed each time a new system is typed. It is normally void, but it can be defined (simply by `\def\everystaff{...}`) to tell MusicTEX to post anything reasonable at the beginning of each system. It was used in the example NWIDOR to post octaviation dashed lines at the end of the piece.

The procedure named `\atnextline`, normally void, is executed at the next computed (through `\autolines`) or forced line break (using `\alaligne` or `\alapage`). More precisely, it is executed after the break and before the next system is typed. Thus it is fit for posting new definitions of layout parameters, when no system is pending[20].

In some scores, tenor parts are not code using the *bass* clef, but using rather the *violin clef* subscripted by a `8`. This is not directly supported by the `\cleftoks`*r*`=` command, but it can be handled using `\everystaff` and `\zcharnote`. As an example the following score

was coded as:

[20]Its logic is similar to plain TEX's `\vadjust` command.

```
\def\nbinstruments{4}
\cleftoksi={6000}
\cleftoksii={0000}
\cleftoksiii={0000}
\cleftoksiv={0000}
\font\eightfig=cmr8
\def\everystaff{\znotes&\zcharnote{-6}{\eightfig
  \kern -1.4\Interligne 8}&\zcharnote{-6}{\eightfig
  \kern -1.4\Interligne 8}\enotes}
\debutextrait
\normal
\NOtes\hu{HIJK}&\hu{efgh}&\hl{hijk}&\hl{hmlk}\enotes
\finextrait
```

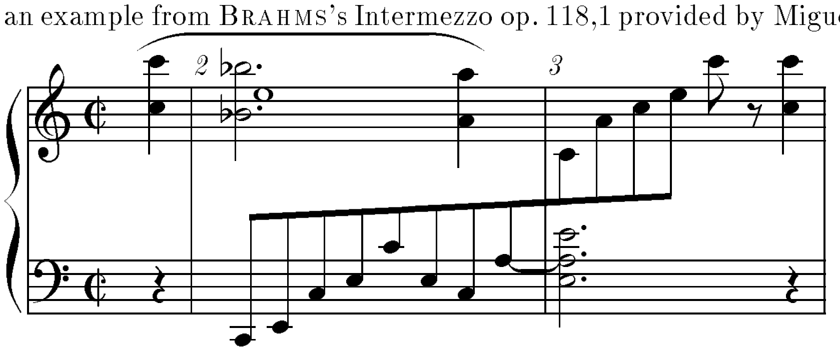
where the `\eightfig` font declaration can obviously be omitted if the TEX format declares an 8 point roman font.

2.17.5 Beams across bars

The `\barre` macro inserts glue and terminates spacing account. Thus it is not compatible with pending *beams* and this is the usual way most composers wrote their scores, at least until the beginning of the XIX-th century. Unfortunately, later composers (Brahms, Scriabin, etc.) wanted to write beams jumping across bars.

This can be partly handled by MusicTEX, using the macro `\xbeambarre`[21] which puts a bar line of global width `\elemskip` with no glue around. Bar counting is done for `\autolines` purpose but no line breaking can occur which mean the beams going across line of page breaks are not supported.

We give an example from Brahms's Intermezzo op. 118,1 provided by Miguel Filgueiras:



whose coding is:

```
\begin{music}
\def\interfacteur{13}
\def\nbinstruments{1}
\nbporteesi=2\relax
\cleftoksi={6000}
\generalmeter{\allabreve}
\debutextrait
```

[21]An ugly mixture of French and English which is therefore not likely to be redefined by somebody else.

```
\normal
\off{1em}\temps\Notes\qp\nextstaff\Ilegu0r\zq{q}\ql{j}\enotes
\barre
\Notes\qsk\ibu0a1\qh0{CEJLcL}\relax
\nextstaff\qsk\rw{l}\pt{p}\zh{_p}\pt{i}\hl{_i}\enotes
\Notes\qh0J\itenl1a\qh0a\nextstaff\tleg0\zq{o}\ql{h}\enotes
\xbeambarre
\Notes\zh{.L.a}\hl{.e}\relax
\nextstaff\tten1\qb0{chj}\tbl0\qb0l\cl{q}\ds\enotes
\Notes\qp\nextstaff\zq{q}\ql{j}\enotes
\finextrait
 \end{music}
```

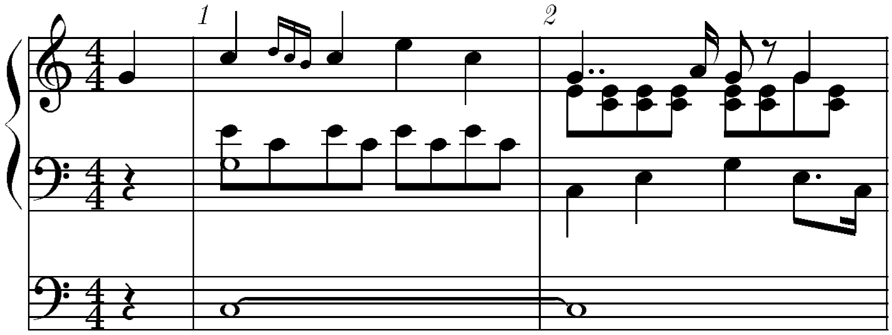
2.18 Small and tiny notes

Before entering details, let us point out that we are presently concerned with typing notes of smaller size than the normal one, without attempting to change the interval between the five lines building a single staff. Changing staff line interval will be treated in a further section.

2.18.1 Cadenzas and explicit ornaments

Ornaments and *cadenzas* usually need to be written using smaller notes[22]. This can be done everywhere by stating `\smallnotesize` or `\tinynotesize`. Normal note size is restored by `\normalnotesize`.

These macros only have a local scope. Thus, if these macros are invoked outside the `\notes...\enotes` pair, the change is valid for the rest of the piece unless explicitely modified but, if they are invoked inside, their effect is local to the current staff of the current `\notes...\enotes` pair. As an example, the following excerpt (beginning of the Aria of the "Creation" by Joseph Haydn)



can be coded as:

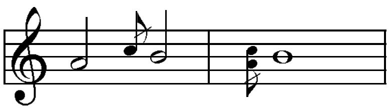
```
\def\DS{\hbox{\ds}}
\def\FS{\hbox{\kern 0.3\noteskip\soupir}\kern -0.3\noteskip}
\def\qbl#1#2#3{\ibl{#1}{#2}{#3}\qb{#1}{#2}}%
\def\qbu#1#2#3{\ibu{#1}{#2}{#3}\qh{#1}{#2}}%
```

[22]This is independent of the staff size.

```
\def\nbinstruments{2}%
\generalmeter{\meterfrac{4}{4}}%
\signaturegenerale{0}%
\nbporteesii=1\relax
\nbporteesii=2\relax
\cleftoksi={{6}{0}{0}{0}}
\cleftoksii={{6}{0}{0}{0}}
\etroit
%
%  end of preliminary definitions
%
\debutextrait
\NOTes\soupir&\soupir|\qu g\enotes
% mesure 1
\advance\barno by -1\relax
\barre\NOtes\itenu2J\wh J&\zw N\ibl0c0\qb0e|\qu j\enotes
\notes&\qbl0c0|\noteskip=0.6\elemskip\tinynotesize
\Ibbu1ki2\qh1{kj}\tqh1i\qsk\enotes
\Notes&\qb0e\tbl0\qb0c|\qu j\enotes
\temps\Notes&\ibl0c0\qb0{ece}\tbl0\qb0c|\ql l\sk\ql j\enotes
% mesure 2
\barre\Notes\tten2\wh J&\ql J\sk\ql L|\ppt g\rlap{\qu g}\qbl1e0\relax
       \zq c\qb1e\zq c\qb1e\relax
       \zq c\tbl1\rlap{\qb1e}\ \ \ccu h\enotes
\temps\Notes&\ql N\sk\pt L\ibl0L{-4}\qb0L|\ibl1e0\zq c\rlap{\qb1e}\cu g\relax
       \zq c\rlap{\qb1e}\raise\Interligne\DS \rlap{\qu g}\qb1g\enotes
\notes&\sk\tbbl0\tbl0\qb0J|\tbl1\zq c\qb1e\enotes
\finextrait
```

2.18.2 Grace notes

Grace notes are a special case of small and tiny notes: the difference is that they are always coded as eighth notes with an oblique bar over the flag. To perform this, special variants of `\cu` and `\cl` have been provided, namely `\grcu` and `\grcl`, with the only difference that the flag has been slashed. Using this together with the note reduction macro, grace notes (optionally chord grace notes) can be easily coded:



The previous example was coded as:

```
\NOtes\qsk\hu h\enotes
\temps\smallnotesize
\notes\grcu j\enotes
\temps\normalnotesize
\NOtes\hu i\enotes
\barre\tinynotesize
```

```
\notes\qsk\zq h\grcl j\enotes
\normalnotesize
\NOTEs\wh i\enotes
```

2.18.3 Gregorian note shapes

Provided that four line staffs are used, *gregorian music* was frequently quoted using specific *neumes*, and this way of coding has been commonly used for the coding of liturgical chant in the Catholic Church until the middle of the twentieth century. Symbols available in MusicTEX are :

- Diamond shaped *punctum*: ◆ = `\diamg` *p* ,
- Square *punctum*: ■ = `\carrg` *p* ,
- Left stemmed *virga*: = `\carpg` *p* ,
- Right stemmed *virga*: = `\carqg` *p* .

They all have a non-spacing variant, which are :

- Non spacing diamond shaped *punctum*: ◆ = `\zdiamg` *p* ,
- Non spacing square *punctum*: ■ = `\zcarrg` *p* ,
- Non spacing left stemmed *virga*: = `\zcarpg` *p* ,
- Non spacing right stemmed *virga*: = `\zcarqg` *p* .

To memorize these symbols, remember that `carr` comes from the French word *carré* which means *square*, that the letter `p` has a tail on the left side and the `q` has a tail on the right side. Other *neumes* can ne obtained by combining two or more of these symbols

2.18.4 Percussion stemless note shapes

Percussion music often uses stemless note heads different from the usual ones.

- White diamond or *rimshot*: ◇ = `\diamw` *p* ,
- The *shaker*: = `\shaker` *p* ,
- The tremolo chord: = `\tremolord` *p* ,

They all have a non-spacing variant[23], which are :

- Non spacing white diamond or *rimshot*: ◇ = `\zdiamw` *p* ,
- Non spacing *shaker*: = `\zshaker` *p* ,
- Non spacing tremolo chord: = `\ztremolord` *p* ,

[23] Stemless, not to be confused with \zq qui extends stems within a chord. For that purpose, see 2.22.3 and 2.18.5.

2.18.5 Other note shapes

The classical note heads given above — namely ● , ○ and **o** — can be replaced with less classical note heads, for example to code special *violin harmonic notes* or *percussion music*. See an example in 2.22.2

At present time, alternate available note heads are :

- The ◆ symbol which is obtained using the `\qu`, `\qh` etc. macros replacing the "q" letter in the macro name with a "y" or writing `\ycu`, `\ycl`, `\yccu`, `\ycup` etc. instead of `\cu`, `\cl`, `\ccu`, `\cup`, etc.
- The ◇ symbol which is obtained using the `\qu`, `\qh` etc. macros replacing the "q" letter in the macro name with a "d" (think of *diamond*) or writing `\dcu`, `\dcl`, `\dccu`, `\dcup`, etc. instead of `\cu`, `\cl`, `\ccu`, `\cup`, etc.
- The + symbol which is obtained using the `\qu`, `\qh` etc. macros replacing the "q" letter in the macro name with a "k" or writing `\kcu`, `\kcl`, `\kccu`, `\kcup`, etc. instead of `\cu`, `\cl`, `\ccu`, `\cup`, etc.
- The × symbol which is obtained using the `\qu`, `\qh` etc. macros replacing the "q" letter in the macro name with a "x" or writing `\xcu`, `\xcl`, `\xccu`, `\xcup` instead of `\cu`, `\cl`, `\ccu`, `\cup`, etc.
- The ⊗ symbol which is obtained using the `\qu`, `\qh` etc. macros replacing the "q" letter in the macro name with the pair "ox" or writing `\oxcu`, `\oxcl`, `\oxccu`, `\oxcup` instead of `\cu`, `\cl`, `\ccu`, `\cup`, etc.

2.19 Staff size

2.19.1 Moving from 20pt to 16pt general staff sizes and conversely

You also want to write some parts of your score in 20pt staff size and in 16pt staff size, namely for distinct parts of pieces. Changing the general staff size is done by saying:

`\musicsize=16\computespecifics` or

`\musicsize=20\computespecifics`

respectively.

2.19.2 Changing staff size for certain instruments

Regardless of the general choice of 16pt or 20pt staff sizes, it is now possible to assign certain instruments – not sperate staffs belonging to a same instrument – to have narrower staff size. This is done by giving a special definition of `\staffspacing`*r* where *r* is the roman numeral of the instrument considered. For example, if the second instrument (starting from the system bottom) is required to have 25% narrower staffs, one only hase to declare:

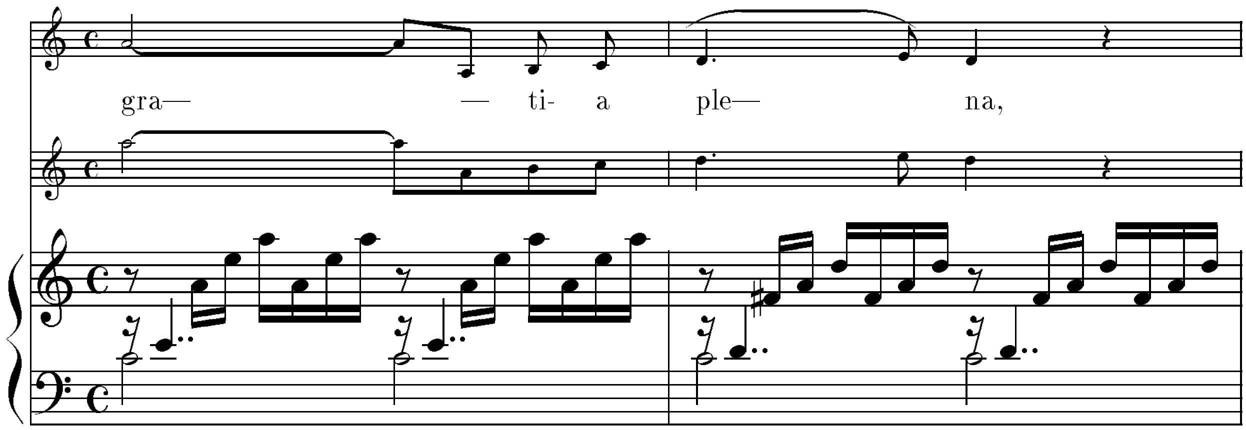
```
\def\staffspacingii{0.75}
```

before the starting command `\debutmorceau`.

Not only does this statement change the vertical spacing between staff lines, but is also changes the size of the key, accidental and note symbols to fit the modified staff line spacing.

However, the existence of discrete font scales for music symbol typing results in some reasonable restrictions of the values of `\staffspacing`r : recommended values are approximately 1, 0.8 and 0.64; if different values are chosen, then the symbol size is taken smaller than the line spacing, which may lead to readable but rather ugly typesetting.

As an example, we give two bars of the *Ave Maria* by Charles GOUNOD using the first prelude of Johann-Sebastian Bach's *Well Tempered Clavier* (transcription for organ, violin and voice, thanks to Markus VEITTES):



This example was coded as:

```
\def\nbinstruments{4}
\sepbarrules
\generalmeter{\meterC}\relax
\def\oct{\advance\transpose by 7\relax}
\def\staffspacingii{0.64}
\def\staffspacingiv{0.64}
\cleftoksi{{6}{0}{0}{0}}
\cleftoksii{{0}{0}{0}{0}}
\cleftoksiv{{0}{0}{0}{0}}
\nbporteesi=2\relax
\nbporteesiii=0\relax
%
\debutextrait
\normal\elemskip=0.02\hsize
%Takt 9
\Notes\zhl c\raise27pt\qs\qupp e|\ds&\oct
  \itenu5h\hl h&gra---&\itenl4h\hu h\enotes
\Notes|\ibbl0j3\qb0h\tqb0l\enotes
\Notes|\ibbl1k0\qb1o\qb1h\qb1l\tqb1o\enotes
\temps\Notes\zhl c\raise27pt\qs\qupp e|\ds&\oct
  \tten5\ibl4c0\qb4h&&\tten4\ibu5g{-3}\qh5h\enotes
\Notes|\ibbl0j3\qb0h\tqb0l&\oct\qb4a&---&\tqh5a\enotes
\Notes|\ibbl1k0\qb1o\qb1h&\oct\qb4b&ti-&\cu b\enotes
\Notes|\qb1l\tqb1o&\oct\tqb4c&a&\cu c\enotes
\barre
```

```
%Takt 10
\Notes\zhl c\raise27pt\qs\qupp d|\ds&\oct
  \qlp d&ple---&\Ilegu4k\pt d\qu d\enotes
\Notes|\ibbu1g3\bigaccid\qh1{^f}\tqh1h\enotes
\Notes|\ibbu2i0\qh2k\qh2f\enotes
\Notes|\qh2h\tqh2k&\oct \cl e&&\tleg4\cu e\enotes
\temps\Notes\zhl c\raise27pt\qs\qupp d|\ds&\oct
  \ql d&na,&\qu d\enotes
\Notes|\ibbu1g3\qh1f\tqh1h\enotes
\Notes|\ibbu2i0\qh2k\qh2f\qh2h\tqh2k&\qp&&\qp\enotes
```

2.20 Layout parameters

Most layout parameters are set by MusicTEX to reasonable default values. However, sophisticated scores[24] may need more place below the lowest staff, between staves, etc. We give below a short list of the most significant parameters.

2.20.1 List of layout parameters

REMARK : the mention "(NOT to be changed)" does not mean that this parameter cannot be changed, but that it should not be modified directly, e.g. by saying something like `\interligne=14pt`. *In other words, changing these parameters must be performed using more comprehensive macros which not only update them but also perform some other compulsory related changes.*

`\interligne` **:** vertical interval between lines in a staff of the current instrument, taking account of a possible specification of `\staffspacing`*r* (see 2.19.2) (*NOT to be changed*)

`\Interligne` **:** vertical distance between the base of staff lines of the current instrument, taking account of a possible specification of `\staffspacing`*r* (`\interligne` is the size of the blank space between lines; the difference between them is the line thickness `\lthick`) (*NOT to be changed*).

`\internote` **:** the vertical spacing of contiguous notes of the current instrument, taking account of a possible specification of `\staffspacing`*r*, i.e. the half of `\Interligne` (*NOT to be changed*)

`\Internote` **:** the vertical spacing of contiguous notes of the instrument(s) whose `\staffspacing`*r* has the *default value* of one (1.0). Since this dimension is the same, regardless of the actual value of the staff line spacing of any instrument, only this value should be used to specify horizontal spacings. It should be used rather than absolute dimensions in `pt` or `mm`, since `\Internote` is duly proportional to the general size (`\musicsize`) but not dependent on specific changes in `\staffspacing`*r* (*NOT to be changed*)

`\nullthick` **:** reserved height above base line for zero staff lines (text of songs)

[24] To our knowledge, the most complicated scores are those written for the piano, during the romantic and post-romantic periods.

`\staffbotmarg` **:** margin below the first staff of the lowest instrument. If not already assigned a non zero dimension, it is set to `\bottomfacteur\Interligne` at the next system.

`\stafftopmarg` **:** margin above the upper staff of the upper instrument.

`\interbeam` **:** vertical distance between beams.

`\interportee` **:** the distance between the bottom of one staff and the bottom of the next one. It is set to 2`\interfacteur\internote` at the next system.

`\Interportee` **:** the distance between the top of one staff and the bottom of the next one. Re-computed at the at the next system.

`\interinstrument` **:** the additional vertical distance between two different instruments. This means that the distance between the upper staff of the previous instrument and the lowest line of the current instrument is equal to `\interportee+\interinstrument`. This value is normally zero, but it helps putting additional space between distinct instruments for the sake of clarity. This is a general dimension which holds for each of the vertical spaces between instruments, except the upper one, in which case this interval is irrelevant. However, this parameter can be overridden for the space above a specific instrument. For example (see the example `angescao.tex`) one can state:

```
\def\interinstrumenti{5pt}
```

to force an additional spacing of 5 points above instrument i, whatever the value of `\interinstrument`. This feature can usefully be used to have more space before instruments representing *voices*, in order to have enough place to put *lyrics* without assigning these lyrics a zero staff specific instrument (useful to avoid having too many declared insruments in a choir score).

`\systemheight` **:** the distance from the bottom of the lowest staff to the top of the highest staff of the upper instrument. This is the height of the vertical bars (single, double, repeats, etc.) (*NOT to be changed*).

In addition, when handling notes of a given staff of a given instrument, the following dimensions are available (note these are not true registers, but *equivalenced symbols* through a `\def`):

- `\altitude` : the altitude of the lowest line of the lowest staff of the current instrument (*NOT to be changed*).
- `\altportee` : the altitude of the lowest line of the current staff (*NOT to be changed*).
- `\stemfactor` : a parameter defining the size of half, quarter and hooked eighth notes stems. Normally a stem has the length of one octave, i.e. 3.5`\Interligne`. However, this is not valid for small size notes and, therefore, the stem size is related to the `\interbeam` dimensions which, in turn, is *normally* equal to 0.75 `\Interligne`. Thus the normal value of `\stemfactor` is 4.66, but it can be shortened for any purpose by saying, for example:

  `\def\stemfactor{3.5}`

  Normal stem length is restored by calling the macro `\normalstems`.

2.20.2 Changing layout parameters

Most of these values can be changed, but only between the end of the previous system and the beginning of the next one. This can be inserted between a `\suspmorceau` (or a `\finmorceau`) and a `\reprmorceau` (or a `\debutmorceau`), but it is wiser to say, for example:

```
\def\atnextline{\global\staffbotmarg=5\Interligne}
```

When doing so, the MusicTEX user should be aware that this could disturb pending *slurs* or *ties*, since the altitude of these is stored in an absolute way, starting from the baseline of the systems. Therefore, changing the `\staffbotmarg` dimension can also be made by means of `\advancebottom{`< *dimension* >`}` which updates all pending slur and tie altitudes by the given dimension. This has been used in `pacifiqn` and `pacifiqb`.

The user may prefer to redefine `\def\bottomfacteur` or `\def\interfacteur` to a given integer number, but this can be done only between a `\suspmorceau` (or a `\finmorceau`) and a `\reprmorceau` (or a `\debutmorceau`), but it is wiser to use `\atnextline` as previously.

It is also wise to use `\atnextline` to change the the number of instruments, the staff spacings, the number of staffs at the next line... provided that the coding of the notes *resists* an unexpected line change executing the `\atnextline`.

In anycase it is a good idea to look at the procedures `\computewidths` and `\computespecifics` to understant what they really compute, and also to find the places where they are actually invoked.

2.20.3 Changing the vertical distance between consecutive systems

If this has to be changed only once at the beginning of a piece, the simplest is to say

```
\def\bottomfacteur{8}
```

before invoking`\debutmorceau`. Note that `\bottomfacteur` is invoked to compute the dimension register `\staffbotmarg` whenever this register contains a zero dimension. The number (it must be a number) is multiplied by `\Interligne` and affected to `\staffbotmarg`.

If this has to be changed several times to meet page layout requirements with scores sometimes – but only sometimes – going very deep below the lowes staff of the systems, it is wiser to use `\advancebottom`, namely:

```
\advancebottom{10pt}
```

to advance the bottom margin register by that dimension at the next system line break.

REMARK : it is also possible to change the system top margin, i.e. `\stafftopmarg` *between systems. Changing top and bottom margin add together but it may influence the position of the whole in the pages.*

2.20.4 Changing staff distance within systems

This can be done at the beginning by changing the definition of `\interfacteur` (it is `\def`ined to a number) before the first `\debutmorceau`. But it is also possible to give a new dimension to the dimension register `\interportee` and, optionally, to the dimension register `\interinstrument` (see above).

2.20.5 Changing the number of lines in staffs

Unless explicitely specified, staffs consist of five lines, in accordance to the normal way of coding music scores. However, two exceptions might be preferred when using MusicTEX :

- *gregorian music* is often writen using staff of four lines instead of five,
- *percussion music* (e.g. drums, triangle) only needs one line staffs, since the pitch cannot change.

Therefore, since its version 5.00, MusicTEX allows for choosing the number of lines of the staffs of an instrument. This is done by defining the macro `\stafflinesnb`*r* — where *r* is the roman numeral of the wanted instrument as usual — to be the number of lines of the specific staff. For example :

```
\def\stafflinesii{4}
```

will make the instrument number 2 (i.e. `ii`) to have staffs of four lines, that is, fit for gregorian music.

2.20.6 Resetting normal layout parameters

Except `\musicsize` which has to be explicitely changed if needed, all layour registers are reset to default values by `\resetfacteurs` which put zero dimensions into `\staffbotmarg` and `\stafftopmarg`, so that the next `\debutmorceau` will recompute them, unless they have been given non zero dimensions inbetween.

2.20.7 Typesetting one-line excerpts rather than larges scores

Very often, what is wanted is not to typeset a large comprehensive score of several lines and pages, but an excerpt of one or two bars, preferably centered such as the various examples of this manual. This can be done simply by replacing `\debutmorceau` with `\debutextrait`[25] and `\finmorceau` or `\suspmorceau` with `\finextrait`.

If you want to terminate it without a bar, you code `\zfinextrait` which acts like `\zsuspmorceau`..

2.20.8 Lyrics

Lyrics can be introduced in two ways. The first one consists in dedicating one instrument whose `\nbportees`*r* is zero. Then the text of the lyrics is just inserted, note by note, by inserting it at the right position between `\notes...\enotes`. The drawback is that a four voice choir needs eight instruments which may lead to troubles if there is also an orchestra score below the voices.

[25] *Extrait* is the French word for *excerpt*.

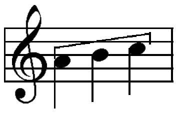
Therefore, another solution consists in adjusting `\interinstrument` and `\staffbotmarg` to give more place below the song instruments[26]. Then the texts of the lyrics are indicated with the song tune instrument, using `\zcharnote` with a negative numeric position value.

By the way, the `\zcharnote` may have a `\vbox` as a second argument, and this `\vbox` may contain several `\hbox`es describing the text of the different couplets (see example `ANGESCAO`).

2.21 Other special ornaments

Since the first release of MusicTEX, many users have either asked for new features of proposed their own procedures to be included in MusicTEX. Some of them have been added because of their high utility, some others have been left aside for several reasons. The major reason for not including many proposals in the MusicTEX release is that – at least on many computers and especially PC's – the TEX memory is limited to 65 000 "words". This is a very restricted value which causes many trials to exceed the capacity of TEX (unless using "bigTEX"). Therefore, only the proposed macros of very general use can be introduced in the release as a standard and the users are suggested to take their specific procedures and include them (directly of by means of an `\input`) in their own source code.

This is particularly valid for people who want to typeset *baroque* music using the ancient *ornament codings* rather than the modern equivalents. As a compromise, some macros kindly provided by Ian Collier in Great Britain have slightly been updated and provided in a separate file named `musicext.tex`.

For example, brackets can be produced:

2.22 Abnormal music coding

2.22.1 Gregorian chant

Gregorian chant is often coded using four line staffs (see section 2.20.5) and using special notes (called *neumes*) which are described in section 2.18.3. But the gregorian chant also needs a special *alto clef* which is in fact the ancester of the modern alto clef. The gregorian *alto clef* can be invoked instead of the modern one by re-declaring the `\clefdut`*r* macro. Thus

```
\def\clefdutiv{\gclefdut}
```

will cause the instrument number 4 (i.e. `iv`) to exhibit a gregorian C clef whenever the value of `\cleftoskiv` refers to an alto clef. The modern alto clef can be restored for that instrument by :

```
\def\clefdutiv{\clefdutsymbol}
```

[26]Beware of a shift of one position, since `\intersinstrument`*r* indicates the interval *above* the *r*-th instrument.

2.22.2 Music score without clefs or with special clefs

Regardless of the number of lines of the staffs, an instrument may have no clefs, e.g. for *percussion music* but also for any weird purpose. This done by declaring some of the following items :

```
\def\clefdesolr{\relax}
\def\clefdutr{\relax}
\def\clefdefar{\relax}
```

At first sight, it could be thought to be silly to cancel the G clef, the C clef and/or the F clef to have an empty clef symbol. But the reason is that the absence of clefs does not mean that notes should not be raised according to their pitch. Thus, if the user uses the G clef coding of pitches, he is welcome to use the present feature to cancel the G clef, but if he uses some alto clef coding, then he should cancel the alto clef symbol. Of course, all this is irrelevant if the engraver chooses to use numeric coding of note position like `\qu{10}`.

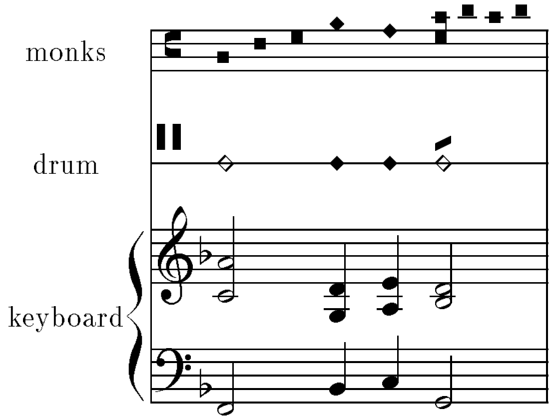
Normal symbols for these clefs and for instrument or roman number r can be restored by :

```
\def\clefdesolr{\clefdesolsymbol}
\def\clefdefar{\clefdefasymbol}
\def\clefdutr{\clefdutsymbol}
```

Besides, a special *drum clef* (two heavy vertical bars) can replace any of the standard clefs, for exemple the G clef by saying :

```
\def\clefdesolr{\drumclefsymbol}
```

It is to be emphasised that these features are specific to one instrument — not one staff of a several staff instrument — so that some weird score for *monks*, *drum* and *electronic keyboard* such as

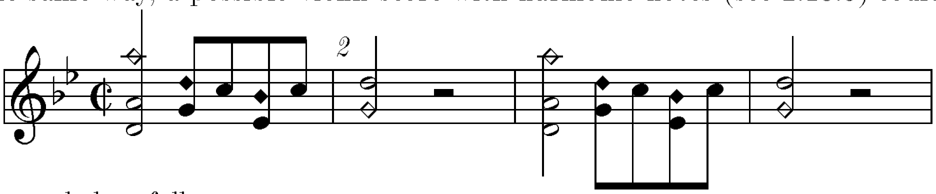


could be coded as follows, regardless this is relevant :

```
\parindent 19mm
\def\nbinstruments{3}   \def\instrumenti{keyboard}
\def\instrumentii{drum} \def\instrumentiii{monks}
\def\stafflinesnbii{1}  \def\stafflinesnbiii{4}
\generalsignature{0}    \generalmeter{\relax}
\signi=-1\relax            % one flat at keyboard
\cleftoksi={6000}\cleftoksiii={3000}
\nbporteesi=2\relax        % 2 staffs at keyboard
```

```
\def\clefdutiii{\gclefdut} % gregorian C clef at instrument iii
\def\clefdesolii{\drumclefsymbol}   % cancel G clef at instrument ii
\def\interinstrumentii{-2\Interligne} % less spacing above drum
\normal
\debutextrait
\Notes\hu F|\zh c\hu h&\diamw O&\carrg {acd}\enotes
\NOtes\qu I|\zq N\qu d&\diamg O&\diamg f\enotes
\NOtes\qu J|\zq a\qu e&\diamg O&\diamg e\enotes
\notes\hu G|\zh b\hu d&\zshaker 3\diamw {OO}&\zcarrg d\carqg g\carrg {hgh}\enotes
\finextrait
```

In the same way, a possible violin score with *harmonic notes* (see 2.18.5) could be :

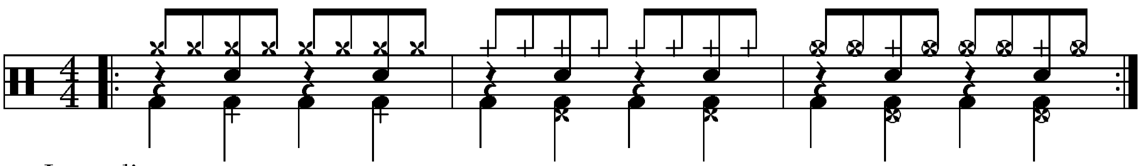


It was coded as follows:

```
\def\nbinstruments{1}\nbporteesi=1\relax
\generalsignature{-2}\generalmeter{\allabreve}\cleftoksi={0000}
\normal
\debutextrait
\NOtes\zd o\zh d\hu h\enotes
\Notes\ibu0k0\zq g\yh0k\qh0j\zq e\yh0i\tbu0\qh0j\enotes
\barre
\NOTes\zd g\hu k\enotes
\NOTes\hpause\enotes
\barre
\NOtes\zd o\zh d\hl h\enotes
\Notes\ibl0b0\zq g\yb0k\qb0j\zq e\yb0i\tbl0\qb0j\enotes
\barre
\NOTes\zd g\hu k\enotes
\NOTes\hpause\enotes
\finextrait
```

2.22.3 Usual percussion music

Besides single percussion scores usually written using one-line staffs, percussion music involving several instruments is often writen on five-line staffs with a *drum clef*, where the instruments are distinguished by the type of the note heads and the apparent pitch of the note on the staff. We give an example — kindly provided by Agusti MARTÍN DOMINGO :



Its coding was :

```
\begin{music}
\def\nbinstruments{1}\generalsignature{0}\def\stafflinesbi{5}
\generalmeter{\meterfrac44}
\cleftoksi={0000}\def\clefdesoli{\drumclefsymbol}
\normal
\debutextrait
\leftrepeat
\Notes\zql f\rlap{\soupir}\ibu0m0\xh0{nn}\enotes
\Notes\zk d\zql f\zq j\xh0n\tbu0\xh0n\enotes
\Notes\zql f\rlap{\soupir}\ibu0m0\xh0{nn}\enotes
\Notes\zk d\zql f\zq j\xh0n\tbu0\xh0n\enotes
\barre
\Notes\zql f\rlap{\soupir}\ibu0m0\kh0{nn}\enotes
\Notes\zx d\zql f\zq j\kh0n\tbu0\kh0n\enotes
\Notes\zql f\rlap{\soupir}\ibu0m0\kh0{nn}\enotes
\Notes\zx d\zql f\zq j\kh0n\tbu0\kh0n\enotes
\barre
\Notes\zql f\rlap{\soupir}\ibu0m0\oxh0{nn}\enotes
\Notes\zox d\zql f\zq j\kh0n\tbu0\oxh0n\enotes
\Notes\zql f\rlap{\soupir}\ibu0m0\oxh0{nn}\enotes
\Notes\zox d\zql f\zq j\kh0n\tbu0\oxh0n\enotes
\setrightrepeat\finextrait
\end{music}
```

To use these different note heads, one must

- either include a specific percussion file namely `musicper.tex` after the usual `\input musictex` or invoke the LaTeX style option `musicper`;
- use special macro names to replace the usual elliptic back note head with either a double sharp sign or with a +. These macros are :
  - `\zx`, `\xu`, `\xup`, `\xupp`, `\xl`, `\xlp`, `\xlpp`, `\xh`, `\xb`, `\zx`, `\xcu`, `\xcup`, `\xcupp`, `\xccu`, `\xcccu`, `\xccccu`, `\xcl`, `\xclp`, `\xclpp`, `\xccl`, `\xcccl` and `\xccccl`, which behave exactly like `\zq`, `\qu`, `\qup`, `\qupp`, `\ql`, `\qlp`, `\qlpp`, `\qh`, `\qb`, `\zq`, `\cu`, `\cup`, `\cupp`, `\ccu`, `\cccu`, `\ccccu`, `\cl`, `\clp`, `\clpp`, `\ccl`, `\cccl` and `\ccccl`, except that the note head is × instead of ● .
  - `\zox`, `\oxu`, `\oxup`, `\oxupp`, `\oxl`, `\oxlp`, `\oxlpp`, `\oxh`, `\oxb`, `\zx`, `\oxcu`, `\oxcup`, `\oxcupp`, `\oxccu`, `\oxcccu`, `\oxccccu`, `\oxcl`, `\oxclp`, `\oxclpp`, `\oxccl`, `\oxcccl` and `\oxccccl`, to get a note head of ⊗ instead of ● .
  - `\zk`, `\ku`, `\kup`, `\kupp`, `\kl`, `\klp`, `\klpp`, `\kh`, `\kb`, `\zk`, `\kcu`, `\kcup`, `\kcupp`, `\kccu`, `\kcccu`, `\kccccu`, `\kcl`, `\kclp`, `\kclpp`, `\kccl`, `\kcccl` and `\kccccl`, to get a note head of + instead of ● .

2.23 Writing your own macros: the `\catcode` problems

As seen before, the `\catcodes` of the `|` and `&` symbols are modified by MusicTEX, in the range of the actual scores but no more[27] in the whole of the TEX source. Thus, if you define your own macros to make your writing easier, you are likely to invoke the `|` or `&` symbols in a part of text where their `\catcode`s are not correctly set. This may result typically in a diagnostic like :

`! Misplaced alignment tab character &.`

when you attempt, not to define, but to use your macro using the `&` symbol to change the instrument. Smart TEXers know that the `\catcode`s are attached to the characters *when they are input* and not when they are used; thus you must be sure that `|` and `&` have the correct MusicTEX `\catcode` when the macro is defined, which may well occur outside the actual score.

It is also worth pointing out that the same problem may occur with other punctuation marks like "`<`", "`>`", "`^`", etc., if their `\catcode` has been changed by some other set of macros, like `french.sty`.

2.24 MusiclaTEX

2.24.1 The `musictex.sty` style

As said before, the amount of memory and registers used by MusicTEX makes it hardly compatible with LATEX. However, Nicolas Brouard succeeded in building a `musictex.sty` which is now included in the distribution. This is not recommended to make separate music scores. Its purpose is rather to provide a means of inserting short musical excerpts in books or articles written with LATEX. Then, the `\documentstyle` command should include `musictex` in the options.

The LATEX style file `musictex.sty` simply `\input`s the following files (in that very order):

- `musicpre.tex`
- `musicnft.tex`
- `musictex.tex`
- `musicvbm.tex`
- `musicpos.tex`

In the case of a LATEX user wanting to use accidental transposition facilities, he should invoke `musictrp` in the options of the `\documentstyle` command.

In case of `TeX capacity exceeded...`, use a "BiglaTEX" (after checking there is no visible error in the source code).

2.24.2 Wide music in LATEX

Another difficulty appears with LATEX: internal LATEX macros handle the page size in a way which is not supposed to be changed within a given document. This means that text horizontal and vertical sizes are somewhat frozen so that one can hardly insert pieces of music of page size different from the size specified by the LaTeX *style*. Although a *largemusic* has been provided,

[27]Since version 4.99.

the main drawback is an unpredictable behaviour of top and bottom printouts, especially page numberings.

If the whole of a document has wide pages, it can be haddled with the `a4wide` style option, or any derivate of it.

2.24.3 The `\catcode` problems

The `musicpos.tex` file merely overrides the `\catcode`s of the `|` and `&` symbols which are modified by MusicTEX. To have access to these symbols when coding music, on should then enclose the scores or excerpts within `\begin{music}` and `\end{music}`. But there is also another possibility, i.e. to say `\nextinstrument` instead of `&` and `\nextstaff` instead of `|`.

Another problem comes from the `french.sty` written by Bernard GAULLE which is the standard of the GUTenberg French association. This style changes many `\catcode`s which lead MusicTEX to fail in many cases. Therefore, since the version 4.99, the `\catcode`s of all are forced to the adequate value at `\debutmorceau`, `\debutextrait` and restored at their original value at `\finmorceau`, `\suspmorceau` and `\finextrait`. This means that some facilities like the *guillemets* or the *tabulation* character are inhibited within music scores (possible problem with sophisticated *lyrics*) but perfectly available within the normal text.

Anyway, in case of emergency, one can invoke `\catcodesmusic` to establish the `\catcode`s at their value fit for music, and `\endcatcodesmusic` to reset them at their external value, for example those chosen by `french.sty`.

2.25 Implementation and restrictions

The macroinstruction file MusicTEX contains approximately 2500 lines of code, that is 80 000 bytes approximately. This requires your score to be compiled by the most extended versions of TEX (65 000 words of working memory). It is therefore wise to set `\tracingsstats` to 2 in order to have an information about the memory used in each page. In desperate situations, we recommend using the "BigTEX" processors which, unfortunately, perform a great deal of disk input/outputs (on PCs with i286 processors) which make them awfully slow[28].

In particular, the number of registers it uses and the amount of memory used by LATEX macros makes it doubtfully compatible with LATEX, unless using BigLATEX.

Other precautions are necessary: beware of end-of-line spaces; they corrupt layout and may cause unwanted line breakings after which music symbols seem to *float* in the air without staffs. To avoid that, it is recommended to use `\relax` rather than `%` at the end of source lines.

[28] Using i386 or i486 processors, this problem disappears with the specific version of `emTeX`

Chapter 3

Installation

As seen before, all the files are available at *anonymous ftp* `rsovax.lps.u-psud.fr` (193.55.39.100) in the directory `[anonymous.musictex]`.

This directory normally contains `musictex.zip` which contains all the distribution for PC (MS-DOS) computers. This is only for `ftp`-ing convenience since all source files are directly available in the same directory. In addition a set of examples (i.e. the files whose name does not begin with "`music`") is packed into `musicexa.zip`, and `recueil.zip` contains the zipped DVIs of a large subset of the examples. Finally the PK files of specific fonts are provided in `musicpk.zip`; getting this file is useless if you are able to `metafont` the files whose `*.mf` are provided in the main package.

The VMS files are also packed into `musictex.bck` . Notwithstanding the fact that files are packed together or not, the files provided are of two kinds:

1. the basic files;
2. the example files.

All *basic files* are either of the form `music*.*` (excluding of course `*.zip` and `*.bck`), `beamn*.*` and `slurn*.*`. Other files (`*.tex` or `*.dvi`) are example files.

Fonts are provided as `*.mf` files but also as `*.tfm` and `*.pk` files for 300 dpi printers or previewers. Additional values of the dpi parameter are also provided in `musicpk.zip`. Normally needed fonts are `musikn20`, `musikn16`, `musikn13`, `musikn11`, `musicbra`, `beamn20`, `beamn16`, `beamn13`, `beamn11`, `slurn20` and `slurn16`.

REMARK : introducing these files in a *format* (with INITEX) is a means of saving computer time.

Chapter 4

Index

Chapter 5

Examples

Due to compatibility problems with LaTeX (used to produce this notice) large examples must be TeX-ed separately, i.e. using TeX and not LaTeX. Therefore, the MusicTeX future user is suggested to produce some of the following examples and to look carefully at the way some special features have been coded.

When producing this examples, care should be taken about the fact that many given files are supposed to be included (by means of `\input`) in other files. Thus the only good files to be directly TeX-ed are those which begin with `\input musicnft`. In addition, it must be noted that most DVI previewers and laser printers have their origin at one inch below and one inch right of the right upper corner of the paper, while the musical examples have their upper left significant corner only at one centimeter right and below the left top of the paper. Therefore, special parameters have to be given to the DVI transcription programs unless special `\hoffset` and `\voffset` TeX commands are introduced within the source TeX text.

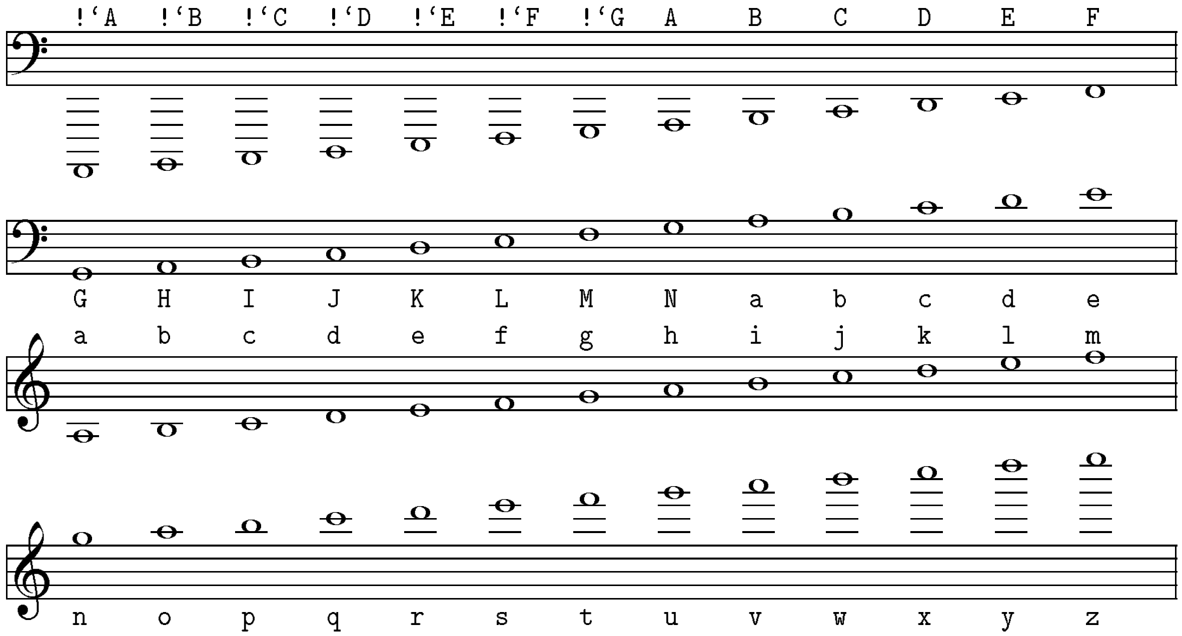
Suggested tests are:

- `PACIFIQN` for a long original piano work (11 pages); `PACIFIQB` is the same in 16 pt staff size;
- `HADAGIO` for a typical organ score (included in `RECUEIL`);
- `CARILLON` for a sophisticated piano score (use BigTeX !);
- `PRAETORI` to get an ancient polyphonic song with three transpositions;
- `ANGESCAO` if you like Christmas carols with four voices, a three-staff organ score and the same transposed to meet ordinary singer's limitations; `ANGESCAM` is the same without organ. See `angescax.tex` to see how the same source has been used for two distinct layouts.
- `HPRELFUG` if you like imitations of J.-S. Bach (included in `RECUEIL`);
- `MARCELLO` if you like pre-baroque music (included in `RECUEIL`);
- `RECUEIL` if you want to get all the organ works of the author in a single book (58 pages);
- `HWIDOR` and `NWIDOR` to get the Toccata by Charles-Marie WIDOR in two different sizes.
- `AVEMARIA` to get the "Méditation" (alias "Ave Maria") by Charles GOUNOD for organ and violin or song.
- `RACINE` to get the "Cantique de Jean Racine" by Gabriel FAURÉ in a transcription fit for organ rather than for piano.

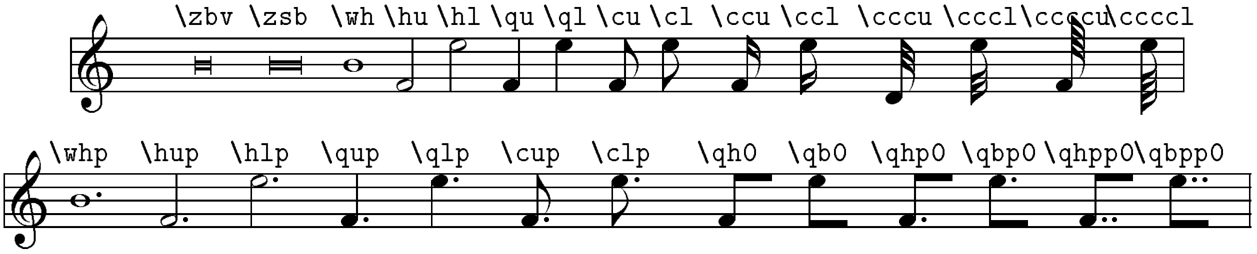
Chapter 6

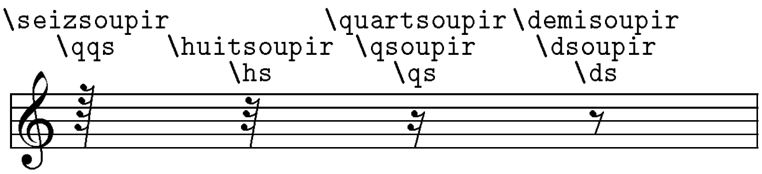
Summary of denotations

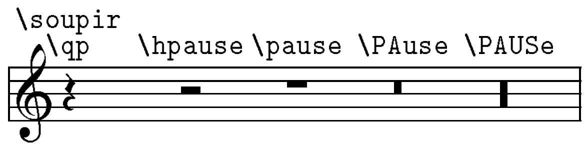
6.1 Pitches



6.2 Notes and Rests







6.3 Other symbols