

# blackmensural.ly

## Black mensural notation in Lilypond 2.12

Lukas Pietsch, January 2011

The `blackmensural.ly` template is designed to support the display of historical polyphonic notation of the late medieval and early Renaissance periods, for purposes such as quoting music snippets in musicological texts and setting mensural incipits in editions of ancient music. While the focus is on getting the display right, it should generally also achieve a correct internal representation of the pitches and rhythms, such that MIDI output will come out correctly.



The periods aimed at include 13th-century *ars antiqua* (Franconian, Petronian) notation, 14th-century *ars nova* and *ars subtilior*, and early 15th-century music written in black notation. It can also be used for later white notation; in that case it will produce a style that looks a bit more like 15th-century manuscripts than like 16th-century prints, as Lilypond's built-in mensural styles do.

The template makes extensive use of embedded postscript for its customized note shapes. It is therefore currently compatible only with Lilypond's Postscript backend, but not the SVG backend. If you need SVG output, you will need an external converter from PS or PDF to SVG. (On Linux, the Evince document viewer can "print to SVG".)

The `blackmensural.ly` software is released under the GNU General Public License. This documentation is released under the GNU Free Documentation License.

### 1. Basic usage

```
\include "blackmensural.ly"
\new BlackMensuralStaff {
  \mensuralTightSetting
  \new BlackMensuralVoice {
    \mensura #'((tempus . #t)(prolatio . #f)) {
      \clavis #'c #3
      \relative c' { c\breve a1 c1 b2 c2 d\breve s\breve }
      \linea "|"
    }
  }
}
```



The customized context types “BlackMensuralStaff” and “BlackMensuralVoice” switch off standard behaviour for beaming, accidentals and bar lines. The `\mensuralTightSetting` command contains settings for horizontal spacing. It should be called right after the `\newBlackMensuralStaff` command. The `\linea` command is just a wrapper around the built-in `\bar` command, used to make a visible bar line.

### 1.1. The `\mensura` command

The `\mensura` command contains the core functionality for mensural notehead shapes and mensural note durations. It takes as its first parameter a Scheme “alist” containing various options, and as its second parameter the block of music to which it is to apply.

The following options may be set in the first parameter:

- `prolatio`, `tempus`, `modus`, `maximodus`: each of these can be set to `#t` (=perfect) or `#f` (=imperfect), defining the mensuration scheme. The default is imperfect for each. Unlike Lilypond’s built-in support for mensural time signatures, the `\mensura` command will actually create the appropriate time measurements and bar structures: provided that at least one of `prolatio` or `tempus` are explicitly set, it will construct bars of one breve containing the appropriate number of four, six or nine minims. If neither `prolatio` nor `tempus` are set, it will instead assume an *ars antiqua* mensuration with perfect *modus*, with a bar length of one maxima consisting of two longae of three breves each. The durations of note values from the semibreve upwards will be automatically adapted to the current mensuration (always assuming perfect durations where the mensuration allows it; context-induced imperfection and alteration must be specified manually.)
- `diminutio`: a diminution factor expressed as an integer, e.g. 2 for *diminutio duplex*
- `sign`: this option controls the display of a mensuration sign. If not set explicitly, a standard mensuration sign (circle or semicircle, with or without dot or vertical line) will be displayed provided that a `tempus/prolatio` bar system is being used; otherwise no sign will be displayed by default. These defaults can be overridden by specifying a sign in several different ways:
  - Boolean value `#f`: suppress display of a mensuration sign
  - a string containing the name of a musicglyph (e.g. “`timesig.mensural32`”)
  - an integer number, to be displayed as a proportion number
  - a pair of two numbers, e.g. ( 2 . 3 ), to be displayed as a proportion
  - an arbitrary scheme expression containing a `stencil` or a markup expression
- `color`: can be set to either `black` or `white`, defining the basic color scheme. Default is `black`. (Divergent coloration of individual notes or note groups can be controlled with the `\coloratio` command; see below)

- `unflagged_sm`: controls the display of noteshapes of semiminims and below. If set to `#f` (the default), a semiminim (quarter note) will be displayed as a rhombic notehead in the same color as the larger notes (depending on the `color` setting), and with a flagged stem. If set to `#t`, it will be displayed in inverted color (hollow if the basic color scheme is black; black if the basic color scheme is white), but without a flag, i.e. as a quarter note in modern notation. The *fusa* and *semifusa* will be adapted accordingly. Note values below the *semifusa* (16th note) are not currently supported.

```
\mensura #'((tempus . #f)(color . black)) {
  \clavis #'c #3
  \relative c' { e\maxima d\longa c\breve b1 a2 a2 e'4 d c d8 e e1 }
  \linea "|"
}
\mensura #'((tempus . #f)(color . white )(unflagged_sm . #t)) {
  \clavis #'c #3
  \relative c' { e\maxima d\longa c\breve b1 a2 a2 e'4 d c b8 a b1 }
  \linea "|"
}
```



- `foldpes`: controls the display of final ascending *longae* in ligatures (*cum perfectione*). If set to `#f` (the default), the final longa will be displayed folded out to the right and with a right downward stem (as in classical white mensural notation). If set to `#t`, it will be displayed folded in to the left, above the preceding note, like a gregorian *podatus* neume (as used in older mensural notation). For more details, see the section on ligatures below.
- parameters for assigning special note stencils to note durations: see final section, *Special note shapes*.

## 1.2. Perfection, Imperfection, Alteration, Coloration

The `\mensura` command will always make each note fill its whole nominal duration in the given mensuration scheme: if `prolatio` is `#t`, every semibreve will have the duration of 3 minims; if `tempus` is `#t`, every breve will have the duration of three semibreves. No special commands exist to mark imperfection or alteration of individual notes. These effects can easily be specified by using Lilypond's built-in compression factor notation, as in `c\breve*2/3` (imperfected breve) or `c1*2` (altered semibreve).

Dotted notes work in the same way as in modern notation (*punctus additionis*). I have not been able to find a good way of encoding division dots (*punctus divisionis*), which look just the same but do not necessarily alter the duration of the preceding note. The workaround is to use a normal dot and then cancel out its lengthening effect by means of a compression factor of  $2/3$ , e.g.: `c\breve.*2/3`.

```
\mensura #'((tempus . #t)) {
  \clavis #'c #3
```

```

\relative c' {
  c\breve           %perfect breve
  b1 c1 d1.*2/3    %group of three semibreves with punct.div.
  e1 d c           %another group of three semibreves
  e\breve          %perfect breve
\linea "|"}

```



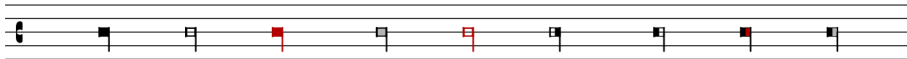
Coloration (as a rhythmic device) can be achieved through combining manual compression factor notation with the `\coloratio` command. Its syntax is:

```

\coloratio #'red { d1*2/3 c b }

```

Supported values for the color parameter are: 'black, 'white (or: 'hollow, 'blackhollow), 'red, 'gray, 'redhollow, 'halfblack, 'halfwhite, 'halfred, 'halfgray. The 'gray option is meant as a convenient replacement for 'red to be used in monochrome print display. The “half-” versions are rare variants needed for some rather esoteric cases (notes reduced to 5/6 of their nominal value).



The `\coloratio` command itself changes only the display but will not affect the note durations, since the required changes are not always strictly predictable. They have to be specified manually in the same way as imperfections and alterations, by using a compression factor such as  $*2/3$ .

### 1.3. Clefs

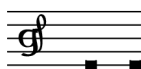
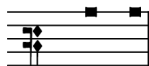
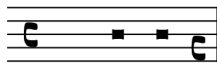
Lilypond’s clef glyphs in the built-in mensural styles (“mensural”, “neomensural” or “petrucci”) don’t fit in well with black mensural notation. The `\clavis` command offers a set of alternative clef glyphs:

```

\clavis #'c #2

```

The first parameter must be #'c, #'f or #'g. The second parameter controls on which line (counted from the bottom) the clef will appear.



BUG: for some reason I don't understand, the new clef glyph sometimes fails to be displayed.

The built-in "Vaticana" style c and f clef glyphs are also suitable for black mensural notation. They can be used with the normal syntax:

```
\clef "vaticana-do2"
```

#### 1.4. Rests

Small rests will be displayed in the built-in "mensural" style by default. For large rests (*longa* and *maxima*) it was necessary to fix Lilypond's buggy treatment of durations: Lilypond thinks that a *longa* rest is always a vertical line two staff spaces long, and a *maxima* rest is either a vertical line of three spaces (in "mensural" style), or alternatively a set of two vertical lines of two spaces each (in "neomensural" style). This is wrong: A *maxima* rest is never a single line, but can consist of either two or (rarely) three parallel lines, each of which can be two or three spaces long, depending on the prevailing *modus* and *maximodus*. A single three-space line is a perfect *longa* rest (= 2 breves), while a two-space line is an imperfect *longa* rest (= 2 breves).

The `\mensura` command will take care of most of these distinctions automatically. If an imperfect *longa* or *maxima* rest is required in a context where it would normally be perfect, this can be specified by `\imperfect r\longa`, or `\imperfect r\maxima`, respectively.

It is often desirable to control for the vertical positioning of rests, for instance in a sequence of several large rests, which are typically set at alternating y-heights. This is done with the `\pausa` command. It takes as its first parameter the line on which the rest is to appear (counted from the bottom), and as its second parameter the rest itself, or a music block containing several rests. If #0 is passed in as the first parameter and the second parameter is a block of several rests, the command will automatically use alternating lines.

```
\mensura #'((modus . #f)) {  
  c'\longa \pausa #2 r\maxima \pausa #1 r\maxima  
  c'\longa \pausa #1 r\longa  
  c'\longa \linea "|"  
}  
\mensura #'((modus . #t)) {  
  c'\longa \pausa #2 r\maxima \pausa #1 r\maxima  
  c'\longa \pausa #1 r\longa
```

```

c'\longa \linea "|"
}
\mensura #'((modus . #f)(maximodus . #t)) {
c'\longa \pausa #2 r\maxima \pausa #1 r\maxima
c'\longa \pausa #1 r\longa
c'\longa \linea "|"
}
\mensura #'((modus . #t)(maximodus . #t)) {
c'\longa \pausa #2 r\maxima \pausa #1 r\maxima
c'\longa \pausa #1 r\longa
c'\longa \linea "|"
}

```



## 1.5. Accidentals

The `blackmensural.ly` template supports the use of the medieval system of accidentals, which consisted not of three but of only two accidental signs: the *b rotundum* or *b molle* (♭), and the *b quadratum* or *b durum* (♮). While the shape of a *b durum* in medieval manuscripts may resemble either a modern natural or a modern sharp, these two functions were not distinguished from each other. While the *b molle* serves to select the lower of two possible alternative values of a note, marking it as a *fa* in the medieval hexachord system (i.e. a note with a semitone step immediately below it), the *b durum* selects the higher of two values, marking the note as a *mi* (with a semitone above). Thus, the same *b durum* sign (♮) is used both for marking a B natural or E natural (as opposed to a flat), and for marking an F sharp or C sharp (as opposed to a natural).

Accidentals specified as part of a key signature, with the `\key` command, work in the normal way. Normally these will not go beyond either one or two flats. The `\mensura` command switches the display of any further *musica ficta* accidentals off by default. If an explicit accidental is required, it must be specified by the command `\accidens`.

```

\relative c' {
  \key f \major
  a1 bes d es d c \accidens b c bes g \accidens fis a\breve
}

```



## 2. Ligatures

I could not find any way of tweaking the built-in mensural-ligature-engraver behaviour so as to allow a black style, so I ended up side-stepping it completely and write all of the ligature behaviour from scratch. The basic command is `\ligatura { ... }`. It supports a somewhat larger number of variant display options than the built-in `\[ ... \]` mechanism for white notation.

Here are the basic binary ligature shapes:

```

\ligatura { b\breve^BL d\longa }
\ligatura { d\breve b\longa }
\linea "|"
\ligatura { b\breve^BB d\breve }
\ligatura { d\breve b\breve }
\linea "|"
\ligatura { b\longa^LB d\breve }
\ligatura { d\longa b\breve }
\linea "|"
\ligatura { b\longa^LL d\longa }
\ligatura { d\longa b\longa }
\linea "|"
\ligatura { b1^SS d1 }
\ligatura { d1 b1 }
\linea "|"

```

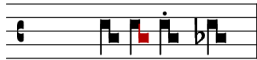


Unlike Lilipond's built-in white ligatures, `\ligatura` also supports colored notes, dotted notes (with the dots printed above the note to which they apply), as well as notes with accidentals (which will be displayed in front of the ligature.)

```

\ligatura { d\breve b\longa }
\ligatura { d\breve \coloratio #'red b\longa } % colored note
\ligatura { d\breve. b\longa } % dotted note
\ligatura { d\breve \accidens bes\longa } % accidental

```



## 2.1. Ligature tweaks

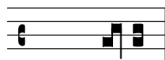
Several commands can be used to tweak ligature behaviour further: `\pes`, `\nopos`, `\obliqua`, `\noobliqua`, `\virga` and `\novirga`.

The `\pes` and `\nopos` commands are used to control whether an ascending note should be folded in to the left, like a *pes* neume in gregorian notation. They can override the standard behaviour set with the `'(foldpes . #t)` option in the `\mensura` command. The folded-in *pes* shapes are the norm for early black notation (until the early 14th century).

```

\ligatura { b\breve d\longa }
\ligatura { b\breve \pes d\longa }

```



```

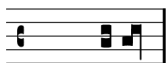
\mensura #'((foldpes . #t)) {
  \relative c' {

```

```

\ligatura { b\breve d\longa }
\ligatura { b\breve \nopres d\longa }
}

```



Similarly, the `\obliqua` and `\noobliqua` commands are used for manually tweaking whether two notes should be displayed as an oblique shape. Either command must be placed before the first of the two notes to which it is to apply. By default, the `\ligatura` command will use oblique shapes automatically in the following cases:

- in a descending ligature *sine perfectione*, i.e. ending in a breve (this behaviour is obligatory and cannot be overridden);
- in a descending ligature *cum opposita proprietate*, i.e. two semibreves (here the oblique shape is the most common, but can be overridden);
- in a ternary or longer ligature, for a descending note pair preceding an ascending final *longa*, if that final note is set to be displayed as a fold-in *pes* (the whole ligature thus taking on the overall shape of a *porrectus* neume in chant notation; this is useful to avoid an ugly clash with the folded-in final notehead.)

```

\mensura #'((foldpes . #t)) {
  \relative c' {
    \ligatura { d1 b1 }           %default obliqua
    \ligatura { \noobliqua d1 b1 } %manually switched off
    \linea "|"
    \ligatura { b1 d1 }           %default no obliqua
    \ligatura { \obliqua b1 d1 }  %manually switched on
    \linea "|"
    \ligatura { c\breve b\breve d\longa } %default obliqua
    \ligatura { \noobliqua c\breve b\breve d\longa } %ugly but possible
    \linea "|"
  }
}

```



The `\virga` command is used to mark that the appearance of vertical stems (*virgae*) on a ligature note should differ from the default behaviour. It can be used in the following cases:

- at the beginning of an ascending ligature *sine proprietate*, it will make the initial stem attach to the left rather than to the right:

```

\ligatura { b\longa d\longa }
\ligatura { \virga b\longa d\longa }

```



- For a medial *longa* or *maxima*, it will turn the stem upwards instead of downwards.

```

\ligatura { c\breve b\longa d\breve }

```



```
\ligatura { c\breve \virga b\longa d\breve }
```



- In the rare cases where a ligature other than one *cum opposita proprietate* ends in an ascending obliqua shape (which must be set manually with `\obliqua`), historical notation theory is ambiguous: both a BB and an LB sequence of this type can appear either with or without an initial downward stem (cf. Apel 1962: 97). The `\ligatura` command will by default assume BB to have a stem and LB to have none (in accordance with Tinctoris); if the opposite values (in accordance with Gafurius) are required, the stem must be removed or added with `\novirga` and `\virga` respectively.

```
\ligatura { \obliqua b\breve d\breve }           ;Tinctoris style
\ligatura { \obliqua b\longa d\breve }
\linea "|"
\ligatura { \obliqua \novirga b\breve d\breve } ;Gafurius style
\ligatura { \obliqua \virga b\longa d\breve }
\linea "|"
```



- For a *maxima* inside a ligature, `\novirga` will remove the default stem

```
\ligatura { c\breve b\maxima d\breve }
\ligatura { c\breve \virga b\maxima d\breve }
\ligatura { c\breve \novirga b\maxima d\breve }
```



## 2.2. Exceptional ligatures

According to Apel (1962: 100), there are occasional non-standard ligatures in early black notation containing a single, non-paired semibreve followed by a longa or breve. These forms are also supported:



### 2.3. Currentes

*Currentes* (also known as *coniuncturae*) are archaic elements of ligature notation, routinely used in the modal ligature notation of the Notre Dame school but still employed occasionally throughout the 13th and early 14th centuries. They are groups of stepwise descending notes at the end of a ligature, which are displayed as a tight row of rhombic noteheads, similar to semibreves. There is no general rule for their rhythmic interpretation. The `\currentes { ... }` command, used inside a `\ligatura { ... }` command, is used to mark a group of notes of any duration as *currentes*. It should normally only be used at the very end of a ligature, and only for stepwise descending notes, but this is not enforced by the program. Ligature notes preceding the *currentes* group will be displayed according to the normal rules, i.e. as if the note preceding the *currentes* was final. It is also possible to have a standalone group of *currentes*, without a preceding normal ligature, or with only a single preceding note, but for technical reasons such groups still need to be wrapped inside a `\ligatura` command.

```
\ligatura { b\breve^BB d\breve \currentes { c1 b1 a\breve } }
\linea "|"
\ligatura { d\breve^B \currentes { c1 b1 a\breve } }
\linea "|"
\ligatura { d\longa^L \currentes { c1 b1 a\breve } }
\linea "|"
\ligatura \currentes { c1 b1 a\breve }
\linea "|"
```



### 2.4. Ligaturae plicatae

A *plica* is a group consisting of a host note (nominally a *longa* or *breve*) followed by a kind of grace note, typically one step above or below the host note. Both are normally displayed together as a single notehead with one or two upwards or downwards stems. The duration of the grace note is typically that of the next shorter value of the host note, and it is subtracted from the host note in such a way that both notes together fill the nominal duration of the host. In modal notation and in very early black mensural notation, *plicae* typically appear at the end of a ligature (*ligatura plicata*). In 14th-century notation, they typically appear as standalone notes.

A *ligatura plicata* can be specified by the `\lplica` command, used in front of the host note and inside a `\ligatura { ... }` command. The host note in these cases must be a *longa*. The subsequent note can be set to any shorter note value. Its duration will be automatically subtracted from that of the *longa*.

The *plica* will normally be displayed in the form of a single upwards or downwards stroke, depending on the relative pitch of the grace note. If the host note is an ascending *longa*, it will be folded out to the right, even if the `foldpes` option is set to `#t`. (Note that the resulting form with a downward *plica* is indistinguishable from the normal form of an ascending final *longa* without a *plica* in the later style, where `foldpes` is `#f`.) An alternative, archaic form can be specified by manually setting the `\pes` option on the *longa* note; it will then be displayed folded in to the left but with an additional smaller notehead with *plica* strokes on the right (cf. Apel 1962: 248f.).

```

\ligatura { d\breve \lplica b\longa c\breve }
\linea "|"
\ligatura { d\breve \lplica b\longa a\breve }
\linea "|"
\ligatura { b\breve \lplica d\longa e\breve }
\linea "|"
\ligatura { b\breve \lplica d\longa c\breve }
\linea "|"
\ligatura { b\breve \lplica \pes d\longa e\breve }
\linea "|"
\ligatura { b\breve \lplica \pes d\longa c\breve }
\linea "|"

```



## 2.5. Standalone plicae

A plica on a standalone longa or breve note outside a ligature is produced by the `\plica` command (which technically, however, is just a thin wrapper around a call to `\ligatura {\lplica ...}`). The plica is displayed as a single square notehead with one or two upwards or downwards strokes of different lengths. Several variant display options (cf. Apel 1962: 334) are supported, each of which can be produced by the `\virga` option.

```

\plica c\longa^L d1
\plica \virga c\longa d1
\linea "|"
\plica c\longa b1
\linea "|"
\plica c\breve^B d1
\plica \virga c\breve d1
\linea "|"
\plica c\breve b1
\plica \virga c\breve b1
\linea "|"

```



## 3. Special note shapes

A large number of special note shapes, so-called *semibreves signatae*, were occasionally used for rhythmic variants of the smallest note values, especially in Italian *trecento* and French *ars subtilior* notation. They typically have the rhombic notehead shape of the semibreve but may carry different combinations of upward or downward stems, flags and other additions, and be colored in different ways.

The Scheme function `make-mensural-note-stencil` can be used to produce customized note shapes of this kind.

The `make-mensural-note-stencil` function takes the following arguments:

- the notehead shape, either 'rhombus or 'quadrata.
- the coloring option, with the same values supported as described for the \coloratio command above.
- the width and height of the notehead (recommended 0.8 1.1 for a semibreve shape)
- an optional list of zero or more additions denoting stems, flags and other modifiers. The following standard elements are currently supported:

- 'stem\_above
- 'stem\_below
- 'flag\_above\_right
- 'double\_flag\_above\_right
- 'triple\_flag\_above\_right
- 'flag\_above\_left
- 'flag\_below\_right
- 'flag\_below\_left
- 'vg\_diagonal (a short diagonal tail on the left)
- 'dovetail\_above
- 'dovetail\_below
- 'pigtail\_above\_right
- 'pigtail\_below\_right
- 'fishhook\_below\_right

In addition to the above symbols, any scheme expression evaluating to a suitable postscript fragment can also be added to the list of modifiers. The `ps_virga` function serves to create such postscript fragments for vertical strokes of arbitrary length, direction and attachment points, mainly suitable for quadratic noteshapes. It takes the following parameters:

- x-offset of the attachment point, measured from the middle of the notehead
- y length (positive for upwards stem, negative for downwards stem)
- coloring option (any of the values described under \coloratio)

Three methods are provided to assign a customized note stencil created with `make-mensural-note-stencil` to some notes:

1. a custom stencil can be assigned to one of the standard note values (*maxima* to *semifusa*), as an option setting in the `\mensura` command. The keys to be used here are the two-letter symbols 'mx, 'lg, 'br, 'sb, 'mn, 'sm, 'fu, 'sf.

```

#(define funnySemiminim
  (make-mensural-note-stencil
    'rhombus 'red 0.8 1.1 'flag_above_left))
\mensura #`((sm . ,funnySemiminim)) {
  \relative c' {
    b2 c4 d4 c1
    \linea "| "
  }}

```



2. In more complex situations, a custom stencil can also be assigned to a fully specified note duration including compression factors, again as an option setting in the `\mensura` command. This method may be useful to implement some of the more complicated notation patterns of Italian *trecento* music, where the normal hierarchy of note values below the breve is largely meaningless.

```

#(define dottedMinimDuration
  (ly:make-duration 1 0 3 2))
#(define ItalianDottedMinim
  (make-mensural-note-stencil
    'rhombus 'black 0.8 1.1 'vg_diagonal))

\mensura #`((,dottedMinimDuration . ,ItalianDottedMinim)) {
  \relative c' {
    b2 c2*3/2 d4
    \linea "| "
  }}

```



3. Finally, the `\signata` command can be used to assign a customized note stencil directly to a note or group of notes. In this case, similar to the `\coloratio` command, the actual note durations must be specified separately, through the appropriate duration values and compression factors.

```

#(define funnyNote
  (make-mensural-note-stencil
    'rhombus 'blackhollow 0.8 1.1 'pigtail_above_right 'stem_below))
#(define strangeNote
  (make-mensural-note-stencil
    'rhombus 'red 0.8 1.1 'fishhook_below_right 'stem_above))
#(define bigNote
  (make-mensural-note-stencil
    'quadrata 'redhollow 1.1 0.75
    (ps_virga 0.55 1.8 'red)))

\new BlackMensuralStaff {
  \mensuralTightSetting
  \new BlackMensuralVoice {
    \mensura #'() {
      \relative c' {
        \signata #funnyNote { d4 e f }
        \signata #strangeNote { d4 }
        \signata #bigNote { d\breve }
        \linea "| "
      }}
  }}}

```



#### 4. References

Apel, Willi (1962): *Die Notation der polyphonen Musik 900–1600*. Wiesbaden / Leipzig: Breitkopf & Härtel.