

musiclyr — a MusiX_{TEX} Extension Package for Lyrics Handling (Version 2.1c)

© Rainer Dunker 1996–2003
rainer.dunker@web.de

June 12, 2003

(The page layout of this document is tailored to A4 paper format, but the text width and height can be freely changed in order to meet other paper sizes; no part of the layout relies on a particular size.)

Contents

1	Introduction	2
1.1	Basic concepts	2
1.2	New since version 2.0	2
1.3	Backward incompatibility issues	3
1.4	Restrictions	3
2	Installation	3
3	Usage	4
3.1	Entering lyrics	4
3.1.1	Lyrics as lines of words	4
3.1.2	Setting up a lyrics line	4
3.1.3	Assigning lyrics lines to staves	5
3.2	Managing the attachment of lyrics to notes	6
3.2.1	Automatic lyrics attachment	6
3.2.2	Addressing manipulations to lyrics	6
3.2.3	Melismas	6
3.2.4	Turning automatic lyrics attachment on/off	7
3.2.5	Lyrics repetition and skipping	8
3.2.6	Miscellaneous	9
3.3	Horizontal positioning	11
3.4	Vertical positioning	12
3.5	Setting lyrics layout parameters	12
3.5.1	Available parameters	12
3.5.2	Individual settings per staff	13
3.6	Multiple verses per staff	13
3.6.1	Handling verses individually	14
3.6.2	Vertical positioning	14
3.7	Auxiliary lyrics	14
3.7.1	Addressing auxiliary lyrics	15
3.7.2	Automatic attachment of auxiliary lyrics	15
3.8	Lyrics at multi-staff instruments	18
3.9	Managing lyrics assignment according to current context	18
3.10	Miscellaneous	19
3.10.1	Avoiding problems after <code>\endpiece... \startpiece</code>	19

3.10.2	Improved centering of words with punctuation marks	19
3.10.3	Verse numbers	19
3.10.4	“Lyrics slurs”	20
3.10.5	Restoring the PMX -unaware behaviour of <code>musixlyr</code> 1.0	20
4	Command index	21
5	Software license	21

1 Introduction

1.1 Basic concepts

`musixlyr` is a set of \TeX macros to be used with Taupin `MusiX \TeX` (version T.52 or later) for typesetting vocal music. Its purpose is to compensate two drawbacks of `MusiX \TeX` ’s lyrics handling:

- Typesetting lyrics with the “native” `MusiX \TeX` commands `\zcharnote`, `\zsong` etc. tends to be quite inefficient, particularly if the lyrics have to be changed or corrected. The idea underlying `musixlyr` is to separate lyrics coding from music coding and let \TeX weave them together with as little manual interference as possible. As a result you can enter and edit lyrics (nearly) as easily as normal text.
- `MusiX \TeX` has no built-in mechanism for centering hyphens between syllables and for handling hyphenation at long melismas. This is implemented in `musixlyr` following the example of engraved music.

1.2 New since version 2.0

`musixlyr` 2.0 provides the following major enhancements:

- Support for lyrics on multi-staff instruments \rightarrow section 3.8
- Melisma declaration as part of the lyrics definition, i. e. within `\setlyrics` \rightarrow section 3.2.3
- Improved scheme for vertical lyrics positioning \rightarrow section 3.4
- Facility for posting lyrics anywhere on a staff, even where no notes are present at all \rightarrow section 3.2.6

These new features are especially interesting when `musixlyr` is used in conjunction with **PMX**:

- Automatic detection of upper/lower voice context in a two-voice staff; explicitly addressing lyrics commands to auxiliary lyrics with `\auxlyr` is no longer required \rightarrow section 3.7.2
- Context-dependent lyrics settings that do not need to be addressed to absolute instrument/staff numbers, making lyrics assignment robust against part extraction \rightarrow section 3.9

Finally, a few improvements have been made that do not bear explicit new features but work rather behind the scenes:

- No more \TeX register allocations for each verse name newly defined by `\setlyrics` or `\copylyrics`; the use of those registers has been replaced by macros.
- Due to the abandoned per-verse register allocations, the `\enableauxlyrics` command is obsolete. It is retained for the sake of backward compatibility, but it has no effect.
- Slightly more robust processing of undefined verses. An error is raised immediately when an undefined verse is assigned to a staff/instrument, and the undefined verse name is implicitly defined as having empty lyrics. This avoids obscure errors in later stages of \TeX processing.

The current version 2.1c is mainly a bugfix version of 2.0, with a few functional additions:

- Possibility to add additional lyrics to an existing lyrics definition without discarding the remainder of the current lyrics line contents; useful if lyrics are defined in several portions spread over the score (see 3.1.2).
- The `musixlyr` version is identified by the predefined macro `\musixlyrversion`, enabling other \TeX code to check for version compatibility (according to a suggestion by Dirk Laurie).

Suggestions for other useful things are welcome.

1.3 Backward incompatibility issues

The following properties make `musixlyr` versions *2.x* incompatible to earlier versions:

- It is no longer possible to represent “empty syllables” by empty \TeX environments ‘{ }’ within `\setlyrics`; see also section 3.1.2.
- Within `\setlyrics`, the underscore character has got a special meaning and thus can no longer be used as usual; see section 3.2.3.
- By default, auxiliary lyrics are now placed vertically *above* (no longer below) the staff they are assigned to; see section 3.7.1.

The following bugfix of `musixlyr` 2.1b also implies a slight backward incompatibility:

- At the beginning of a music line (system), lyrics have often been right-shifted because `musixlyr` has assumed MusiX \TeX ’s horizontal “zero” position as the left boundary for lyrics positioning. Now, lyrics are allowed to extend into the left margin as far as required for centered placement under/above the note.

Although this is a more “natural” behaviour, it may lead to collisions between lyrics and other stuff (like dynamic marks) that have not occurred using older `musixlyr` versions. In such a case, there are two ways to solve the problem:

- Restore the former `musixlyr` behaviour globally by saying


```
\oldlyrlinestart
```

 near the top of your score source.
- Right-shift the colliding lyrics where necessary using `\lyroffset` (see 3.2.6).

1.4 Restrictions

- `musixlyr` does not influence the MusiX \TeX /`musixflx` spacing algorithm in order to make the notes spacing dependent on the space requirements of the lyrics. See section 3.3 about how spacing problems are handled.
- `musixlyr` seems to work correctly in my preferred working environment, but I cannot exclude problems in situations I have not undergone (especially concerning the use of MusiX \TeX preprocessors, the MusiX \TeX extension library, and past or future MusiX \TeX releases). Thus it is important that you point out encountered bugs and incompatibilities to me; there is little chance that I notice them on my own.

Please do also bother me if you find this manual — or parts of it — incomprehensible or incomplete.

2 Installation

Put the `musixlyr.tex` file into a directory that is included in your `TEXINPUTS` search path — that’s all. (Refer to your local \TeX documentation for details on this.)

3 Usage

Before using `musixlyr` macros, state

```
\input musixlyr
```

near the beginning of your source file but **after** the `\input musixtex` (L^AT_EX: `\usepackage{musixtex}`) command.

3.1 Entering lyrics

3.1.1 Lyrics as lines of words

As opposed to the “vertical” music construction of M_us_iX_TE_X, `musixlyr` treats lyrics “horizontally” as *lyrics lines*. Each separate “thread” of words is a separate lyrics line; thus, in the following example¹,

The image shows a musical score with four staves. The first staff is a treble clef with a key signature of one sharp (F#) and a 3/2 time signature. It contains a melody of quarter and eighth notes. Below the first staff are four lines of lyrics, each aligned with the notes above. The lyrics are: "Quand je bois du vin clai - ret, a - mi tout", "Le bon vin nous", "Bu - - vons bien, bu -", and "Bu - - vons bien, bu -". The hyphens indicate syllables split across notes.

there are four lyrics lines notwithstanding that two of them consist of the same words.

Note that the number of lyrics lines does not depend on the number of staves: Any number of lyrics lines (including none) can be attached to each staff (more about multiple lyrics lines per staff in section 3.6).

3.1.2 Setting up a lyrics line

A lyrics line is set up using the command

```
\setlyrics{⟨name⟩}{⟨words⟩}
```

where `⟨name⟩` is a name² for the lyrics line (e. g. the name of the voice singing these lyrics) and `⟨words⟩` is the complete text where multi-syllable words must be given **hyphenated**. For example, the soprano lyrics of the above example are set as follows:

```
\setlyrics{soprano}{Quand je bois du vin clai-ret, a-mi tout}
```

Important notes on using `\setlyrics`:

- Within the lyrics definition, spaces and hyphens are used to split the whole text into words or syllables. If you want a space or hyphen appear *within* a word/syllable, use the command `\space` or `\lyrhyphenchar`, respectively, instead.
- If you want the lyrics attachment to skip over a note, i. e. a certain note shall not receive a syllable, you may achieve this by inserting an empty syllable in the `⟨words⟩` text. However, neither a “really empty” syllable (i. e. two consecutive separating spaces) nor an empty environment `{}` will work, but you may use either `{\empty}` or `\empty{}` for this.

¹Excerpt from “Tourdion” by PIERRE ATTAIGNANT (*ars musica* vol. 4, Möseler Verlag Wolfenbüttel/Zürich)

²This name must not contain backslashes or commas. Most other characters do not cause problems; however, it is safer to use only letters and digits.

Normally, the `\nolyr` command (discussed later in this manual) should be used for this purpose, but an empty syllable may be more handy if it occurs in only one of multiple assigned verses; see also sections 3.2.3 and 3.6.

- Note that, since lyrics lines are cut into pieces which are processed separately, you cannot apply things like temporary font changes in the usual manner (i. e. using `\TeX` environments). See section 3.5 about how to accomplish layout changes.
- Applying `\setlyrics` to an already defined `\langle name \rangle` redefines the lyrics line. This may be useful at very long pieces because setting up and applying lyrics in smaller portions (e. g. movement by movement) saves both time and memory.
- It is possible to specify melismas — that shall occur later when a verse is attached to the music — already within `\setlyrics`; see section 3.2.3 for details.

There are two variants of the `\setlyrics` command available:

- If several voices have the same lyrics, you do not have to copy and paste the lyrics for each further voice; just say

```
\copylyrics{\langle from name \rangle}{\langle to name \rangle}
```

to set up a lyrics line named `\langle to name \rangle` with the same contents as `\langle from name \rangle`.

- Applying `\setlyrics` to an already defined `\langle name \rangle` has the effect of discarding all previous contents of `\langle name \rangle`, even if part of the lyrics is still pending for being processed. If you want to *append* further lyrics to an already existing lyrics line instead — without interrupting the processing of the current lyrics contents —, use the command

```
\appendlyrics{\langle name \rangle}{\langle words \rangle}
```

instead of `\setlyrics`.

3.1.3 Assigning lyrics lines to staves

Note: During the following discussion, the term *staff* is used wherever MusiX`\TeX` would precisely use the term *instrument* instead, assuming that lyrics are posted to single-staff instruments only. However, everything also applies to the assignment of lyrics to staves of multi-staff instruments; see section 3.8 about how to use lyrics in a multi-staff instrument context.

To assign a lyrics line to a staff, say

```
\assignlyrics{\langle i \rangle}{\langle name \rangle}
```

where `\langle i \rangle` is the instrument number (counted as usual from bottom to top) and `\langle name \rangle` the name of the lyrics line.

The resulting assignment can be redefined everywhere. This is useful if the arrangement of voices on staves changes within the score; when a voice is continued on a different staff at the next system, use `\assignlyrics` (preferably enclosed in `\atnextline{...}`) to make the respective lyrics line follow this change without interrupting the flow of words.

If a lyrics line is assigned to instrument `\langle i \rangle` and you want it to be continued on a different instrument while assigning no other lyrics line to instrument `\langle i \rangle`, you must explicitly cancel the former assignment by saying `\assignlyrics{\langle i \rangle}{}`.

Caution: Never assign a lyrics line to more than one staff simultaneously; if the same lyrics are needed at several staves in parallel, use `\copylyrics` (see above).

As a summary of this section, here is the complete “lyrics definition section” of the above example:

```

\setlyrics {soprano}{Quand je bois du vin clai-ret, a-mi tout}
\setlyrics {alto}   {Le bon vin nous}
\setlyrics {tenor}  {Bu-vons bien, bu}
\copylyrics{tenor}  {bass}
\assignlyrics4{soprano}
\assignlyrics3{alto}
\assignlyrics2{tenor}
\assignlyrics1{bass}

```

See also section 3.9 about an alternative way to assign lyrics to staves.

3.2 Managing the attachment of lyrics to notes

3.2.1 Automatic lyrics attachment

By default, the syllables of a lyrics line are consecutively attached to all *spacing* notes of the assigned staff. Chords thus receive only one syllable since they usually contain exactly one spacing note. However, this may be different if you are using auxiliary lyrics; see section 3.7 for details.

“Spacing” in this context means: generated by one of the commands `\qu`, `\hu`, `\wh`, etc. and not by `\zqu`, `\zhu`, `\zq`, etc. Spacing or not does *not* depend on whether the note-generating command is enclosed in `\notes` or any other `\notes` command. So if any note seems to have been left out by the attachment automatism, first make sure that this note is generated using a command not beginning with `\z`.

If the number of words of a lyrics line gets exhausted before all notes of the respective staff are processed, “???” is posted at all remaining notes.

3.2.2 Addressing manipulations to lyrics

In the “ideal” case, the attachment automatism perfectly distributes the lyrics over the music. In practice, however, situations in which the automatism must be “helped” manually are frequent. Such manipulations must be posted within the music code. Here are the basic rules:

1. All lyrics manipulations must be stated inside `\notes... \enotes`. (Exception: melisma specifications within `\setlyrics`, see section 3.2.3).
2. Lyrics manipulations solely refer to the lyrics line(s) attached to the current staff (“current” with respect to the enclosing ‘&’ characters).
3. Similar to many other MusiX_{TEX} commands (e. g. for beams and slurs), lyrics manipulations must be issued before the corresponding note commands.

Examples in the following sections will illustrate the application.

3.2.3 Melismas

Regarding lyrics and music separately, melismas are no property of either one; instead, they characterize the relation between both and play a role only where music and lyrics occur together. Using `musixlyr`, however, lyrics coding and music coding are basically unrelated, but melismas do have to be specified somewhere. You have to decide yourself whether you regard melismas as an attribute rather of the lyrics or rather of the music (this may vary according to the type of score you are preparing, or may even differ within a score); `musixlyr` provides two corresponding styles of melisma specification which are discussed in the following paragraphs.

Melismas as a music property. Within the music code, a melisma is initiated and terminated by the command pair

```

\beginmel
\endmel

```

issued before the first and last note, respectively, of the melisma. This left-justifies the current word/syllable, suspends lyrics until after the melisma, and sets up a “word extension underline”³ unless the current word is continued afterwards and thus has to be hyphenated. For example,



has been coded as:

```
...
\Notes\beginmel\qa{def}\endmel\qa e\en\bar
\Notes\beginmel\qa{def}\endmel\qa g\en\bar
\Notes\beginmel\qa{'a!gf}\endmel\qa e\en\bar
\Notes\beginmel\qa{defe}\en\bar
\NOTEs\endmel\wh d\en
...
```

If a system break occurs within the melisma, hyphenation or underline is continued at the next system, thus automatic line breaking is applicable.

Melismas as a lyrics property. To specify a melisma within a lyrics definition (i.e. within `\setlyrics` or `\appendlyrics`), simply append an underscore character to a syllable to make it span over an additional note, two underscores to span over two additional notes, and so on. Alternatively, you may append to a syllable a single underscore followed by a number which denotes the number of *additional* melisma notes: `abc_` is equivalent to `abc_1`, `abc_` to `abc_2`, and so on.

No matter which of the two underscore appendix styles you use: If a melismatic syllable is hyphenated to the following one, the melisma denotation must **precede** the hyphen — for example: `abc_--def` (**not**: `abc--_def`).

All melisma properties mentioned above (lyrics suspension, left justification of syllable, word extension underline, system break support) also apply to melismas specified this way. For reference, here is the lyrics definition section of the example above, this time utilizing the underscore melisma technique (note that, in this case, all `\beginmel` and `\endmel` commands must be omitted in the music section):

```
\setlyrics{melismatic}{a_ me_3-lis_--ma_4}
```

Comparing both points of view. The music-oriented melisma view (using `\beginmel`/`\endmel`) is advantageous for music with a rather loose binding to certain lyrics, e.g. folk songs that are to be printed with various verse compilations or in different languages. You may then enter each verse without taking care of melismas because they are coded just once as part of the music.

On the other hand, the lyrics-focused melisma style (using underscores) is best suited to pieces with a fixed lyrics-to-music binding; it is easier to type and to read, but less flexible as regards combining the same lyrics with different music. It is especially advantageous for multiple verses (see section 3.6) with different syllables-to-notes distribution (see verse 3 of the “Drunken sailor” example in section 3.10.3); it is much easier to code each verse’s melismas individually than tracking all melismas in parallel using the `\verses` command (see again section 3.6).

Note. You may use both styles within the same score but you should avoid activating both melisma types simultaneously at the same lyrics line; the result of such an occurrence is undefined.

3.2.4 Turning automatic lyrics attachment on/off

If the attachment automatism is unsuitable for some situation — e.g. at complex polyphonic voicings or wherever you want to place all syllables manually — you can turn it off using

³I am not sure whether this is the appropriate term; it is taken from the *Finale* reference manual.

`\lyricsoff`

after the last note that shall get a syllable. (If your piece begins with a lyrics-less section, state `\lyricsoff` before the first note of a staff.) To re-enable automatic lyrics attachment, use

`\lyricson`

before the first note that shall get a syllable.

If automatic lyrics attachment shall skip over certain single notes only, the `\nolyr` command is more suitable (see section 3.2.6).

3.2.5 Lyrics repetition and skipping

As in the following example of a polyphonic piece,



it often happens that some lyrics phrases are repeated by some but not all voices, or that each voice has its individual arrangement of lyrics repetitions. Nevertheless, you do not have to enter these repetitions directly within `\setlyrics`; instead, you enter the lyrics as if they were performed in an unbroken sequence. Repetitions and skips are then handled using a “go to” technique involving the following steps:

Labelling “go to” targets. A syllable which has to be jumped at must be preceded (without spaces in between) by

`\llabel{<label name>}`

within `\setlyrics`.

Performing jumps. Within the music code, say

`\golyr{<label name>}`

before the note at which the jump has to be performed.

Caution: If you try to jump to an undefined lyrics label, no meaningful error message is produced. However, if \TeX reports an error at a `\golyr` command, you should check conformity of defined and used labels.

The following actions must be taken in special cases:

Adding punctuation marks. Usually a comma or period is appended to the word preceding a jump; this is done by the command

`\lyrpt<punctuation char>`

where `<punctuation char>` can be any single punctuation character.

Removing punctuation marks. Before adding a punctuation mark to a word as said above, an already existing punctuation mark should be removed by the command:

`\lyrnop`

Lower case letters. Words at jump targets often begin with a capital letter which should appear lower case after a jump; this can be forced by the command:


```
\lclyr
```

Here is a code excerpt from the above example, summarizing the use of all mentioned commands:

```
\setlyrics{amen1}
    {\llabel{amenlabel}A-men!} \assignlyrics1{amen1}
\copylyrics{amen1}{amen2}      \assignlyrics2{amen2}
\startextract
\NOTes% lower staff:
    \itenu0K\beginmel\wh K\sk\sk\sk
    \midslur2\tten0\endmel\hlp K&%
% upper staff:
    \qa f\lyrnop\lyrpt,\qa g%
    \golyr{amenlabel}\lclyr\qa h\lyrnop\lyrpt,\qaf_i)%
    \golyr{amenlabel}\lclyr\beginmel\islurdlh\qa{hg}%
    \tslurif\endmel\qa f\en
\NOTEs\wh N&\wh g\en
\endextract
```

Note that the “go to” technique is recommendable mainly if different voices perform different repetitions or skips on the same lyrics; if all voices perform the same sequence of repetitions, however, it is easier to hard-code them within `\setlyrics`.

3.2.6 Miscellaneous

Excluding single notes. In the following situation,



the eighth note must be excluded from receiving a syllable. This can be accomplished saying

```
\nolyr
```

before the command generating this note.

Manual lyrics posting. When automatic lyrics attaching is switched off, a word or syllable can be placed manually saying

```
\lyr
```

before the command generating the note that shall get this word/syllable.

This command also works independently from a following note command; you can issue it before a rest, a `\sk` or even alone for itself in order to place a word/syllable wherever you want (no matter whether the attachment automatism is on or off).

Manipulations to the posted word/syllable have to be placed *before* the `\lyr` command.

Left justification. Say

```
\llyr
```

in order to have the next word/syllable left-justified to the notehead instead of centered.

If you prefer to have left justification as the default, say

```
\leftlyrtrue
```

before your piece begins (`\leftlyrfalse` to restore centered lyrics).

Word extension underlines. The melisma commands (see 3.2.3) automatically manage word extension underlines; however, they can be set manually: Say

`\lyrrule`

before the respective note command to initiate an underline *after* the word, and

`\lyrruleend`

before the note where the underline shall be terminated.

Note that no underline will be drawn if the word is continued afterwards so that the underline would interfere with the hyphenation.

Horizontal offset. To shift a word/syllable out of its default horizontal position, say

`\lyroffset{<n>}`

which — similar to `\roffset` — moves the next word/syllable `<n>` quarter note head widths to the right (left if `<n>` is negative).

Note that the effect of `\lyroffset` may be overridden by automatic lyrics shifting. If you want to gain complete manual control over the positioning, you have to disable the shifting automatism; see section 3.3 for details.

Extra lyrics. If, as in the following example⁴,

The image shows a musical score for two staves. The top staff is in treble clef with a key signature of one sharp (F#). The bottom staff is in bass clef with a key signature of one sharp (F#). The melody consists of quarter notes and eighth notes. The lyrics are: 'yel - low sub - ma - rine,' on the first line and 'yel - low sub - ma - rine, ya, ya, ya,' on the second line. The 'ya, ya, ya' part is positioned below the notes, indicating it is an extra lyric.

a voice has “extra” lyrics (such as “hmm”, “ah”, “ya”, etc.) in addition to the “main” lyrics, you need not enter them within `\setlyrics` (which would inhibit using `\copylyrics`); instead, you can enter them directly within the music code using the command

`\lyric{<extra syllable>}`

before the note where they shall appear. (Note that the effect is different from using `\csong` which would not apply automatic hyphenation, horizontal positioning control, and `musixlyr`’s vertical positioning scheme.) Automatic lyrics attaching — if active — is suspended at that note. All manipulations can be applied; they must be given before the `\lyric` command.

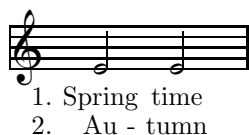
If you want the syllable linked to the following one by a hyphen, use `\lyrich` instead of `\lyric`.

It may sometimes be necessary to post lyrics where no note is present at all, i. e. in the empty space between notes. This is, however, possible with `\lyric`, but is unsuitable because automatic lyrics attachment shall *not* be suspended at the next note. In this case one of the commands `\lyric*` or — for hyphenated syllables — `\lyrich*` should rather be used; they have no effect on the “regular” flow of lyrics. One application for this are vertically aligned verse numbers. For example,

```
\setlyrics1{Spring time}
\setlyrics2{Au-tumn}
\assignlyrics1{1,2}
...
\znotes\loffset{2.2}{\verses{\lyric*{1.},\lyric*{2.}}}\en
\hardspace{4mm}%
\NOTes\hu{ee}\en
...
```

⁴Excerpt from “In the town where I was born” (*Eine kleine Melodie*, Strube, München 1989)

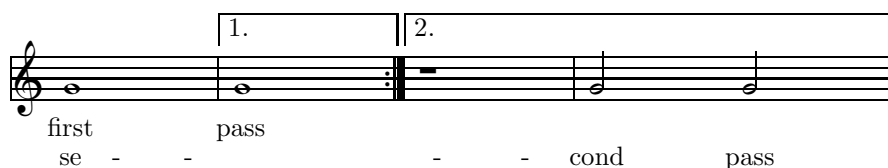
yields:



(See section 3.6 about attaching multiple verses to a single staff, a feature that has not been introduced so far.)

You may even obtain verse numbers that are repeated at the beginning of each system by enclosing the `\znotes...` command for setting the numbers in `\def\everystaff{...}`.

Another possible application — combined with empty syllables — is coping with unusual lyrics situations at voltas; for example,



is accomplished by:

```
\setlyrics1{first pass}
\setlyrics2{se-cond pass}
\assignlyrics1{1,2}
\startextract
\NOTES\wh g\en
\Setvolta1\def\atnextbar{\znotes\verses{,\lyric{}}\en}\bar
\NOTES\wh g\en
\znotes\verses{,\lyrich*{}}\en
\Setvolta2\setendvoltage\rightrepeat
\NOTES\verses\lyricsoff\pause\en\bar
\NOTES\ha{gg}\en\setendvolta
\endextract
```

The trick used here is to inject empty syllables into the second verse at the barlines around the first volta; they serve as anchors for the hyphenation suspended under this volta.

Note that the `\lyric`, `\lyric*`, `\lyrich`, and `\lyrich*` commands cannot be used at staves without assigned lyrics lines.

3.3 Horizontal positioning

`musixlyr` takes care of blank space between words. If a word — at its default position — would come too close to or overlap the previous word, it is shifted right in order to ensure a minimum of blank space. This minimal interword space can be set by

```
\minlyrspace=<any TEX dimension>
```

and is initially set to 3 pt.

The situation is different if two syllables belong to the same word. Such syllables are allowed to close up; only overlapping is inhibited by shifting. If the “natural” space between such syllables is narrower than a hyphen, the hyphen is omitted and the two syllables close up completely (eventually by left-shifting the second syllable to prevent a small gap). You may though force hyphens in every spatial situation saying

`\forcelyrhyphenstrue`

(to restore the default behaviour: `\forcelyrhyphenfalse`). In this case, enough room is made for the hyphen by right-shifting the second syllable if necessary.

If the note spacing is quite narrow, lyrics shifting may get accumulated within a system. In order to prevent shifting words too far out of their original position, a wider note spacing should be considered. The command

`\showlyrshiftrue`

marks each lyric shift with a black rectangle of the respective width, thus making it easy to detect questionable positionings.

If you prefer to completely disable lyrics shifting, set `\minlyrspace` to a large negative value (e.g. -10 cm).

3.4 Vertical positioning

By default, `musixlyr` places lyrics vertically centered between the two adjacent staves. Lyrics below the lowest or above the topmost staff of a system are centered around the system's bottom or top margin line.⁵

In most scores, you will have to deviate from this standard and manually adjust the lyrics' vertical position. This can be done per staff with the command

`\lyrraise{<n>}{<pos>}<raise>`

which applies the following setting to the lyrics attached to staff number `<n>`: `<pos>` is one of the letters `a` and `b`. `a` sets the lyrics above the staff, `b` below the staff. `<raise>` is a dimension value by which the lyrics are lifted above (or lowered under, if it is negative) the default position as mentioned above. `\lyrraise{<n>}{bOpt}` is the initial setting for all staves.

It is possible to apply changes to the vertical position at any place within the score, but in most cases it is most reasonable to do so within `\atnextline{...}` which ensures consistent, even spacing within each system.

For the sake of backward compatibility, applying MusiX_{TEX}'s command `\setsongraise{<n>}{<raise>}` implicitly performs a `\lyrraise{<n>}{b<raise>}`, but not vice versa — i.e. `\lyrraise` has no effect on lyrics set with MusiX_{TEX}'s lyrics commands `\csong`, `\zsong`, etc. Likewise, the obsolete `musixlyr` command `\auxsetsongraise` implies `\auxlyr{\lyrraise...{b...}}`, but not vice versa (see section 3.7 to learn about the meaning of `\auxlyr`).

3.5 Setting lyrics layout parameters

3.5.1 Available parameters

Font: There is no special font command; the current font is used for lyrics. It can be changed using the usual \TeX / \LaTeX commands.

Minimum length of word extension underlines: A word extension underline is omitted if it would become too short. The minimum length can be set by

`\minlyrrulelength=<any \TeX dimension>`

and is initially set to 2mm. If you prefer to completely inhibit the underlines, set this parameter to an unsurpassably high value (e.g. the width of your paper).

Minimum inter-syllable distance for multiple hyphens: If the space between two hyphenated syllables (or between one of the syllables and the respective system margin) is wider than this distance, it is filled with a line of hyphens instead of a single hyphen. The minimum distance to be filled with more than one hyphen can be set by

⁵This is not necessarily the most reasonable way, but it is compatible to MusiX_{TEX}'s `\csong` behaviour.

```
\minmulthyphens=(any TEX dimension)
```

and is initially set to 15mm. If you prefer to completely inhibit filling with multiple hyphens, set this parameter to an unsurpassably high value.

Hyphen character: This can be redefined by:

```
\def\lyrhyphenchar{...}
```

Further layout parameters discussed elsewhere are: vertical positioning (section 3.4), word justification (section 3.2.6), minimum interword distance, and forcing hyphens (section 3.3).

Note: In order to change the mentioned parameters globally, the respective commands must be issued *outside* `\notes... \enotes`.

3.5.2 Individual settings per staff

All layout parameters can be set either globally (applying to all lyrics lines) or individually, i. e. valid for a single specified staff. An individual change is accomplished by enclosing the proper command in

```
\lyrlayout{...}
```

and issuing this within `\notes... \enotes` in the section of the referred-to staff. For example,

```
\notes...&\lyrlayout{\it\leftlyrtrue}...&... \en
```

applies italic font and left justification to the lyrics line(s) assigned to the second staff.

Applying `\lyrlayout` to the same staff again overrides *all* previous individual settings for this staff; pay attention to this if you have made several individual settings and want to reset some but retain others. Saying `\lyrlayout{}` restores the global settings for a staff.

The `\lyrlayout` command can also be used within `\setlyrics`; this is useful if an individual setting (especially a font change) should apply to a particular portion of lyrics rather than to a portion of music. For example,

```
\setlyrics{x}{An \lyrlayout{\it}em-pha-sized\lyrlayout{} phrase}
```

causes the word “emphasized” to appear italic wherever it is posted within the music. (Be sure to attach `\lyrlayout` commands to the corresponding lyrics portions *without* blank spaces in between.)

As an exception, the **vertical positioning of lyrics** cannot be changed within `\lyrlayout`; instead, it is handled staff-wise using the special commands discussed in section 3.4.

3.6 Multiple verses per staff

An arbitrary number of verses can be assigned to a staff straightforwardly by saying

```
\assignlyrics{<n>}{<names>}
```

where `<names>` is a **comma-separated list** of the desired lyrics line names, listed from top to bottom. In the following example⁶,

Non mo - ri - - - ar ___ sed
Ich wer - de nicht ___ ster - ben, son - dern

the lyrics lines have been set up as follows:

```
\setlyrics{latin} {Non mo-ri-ar sed}
\setlyrics{german}{Ich wer-de nicht ster-ben, son-dern}
\assignlyrics1{latin,german}
```

⁶Excerpt from: “Non moriar sed vivam”, motet by MARTIN LUTHER

3.6.1 Handling verses individually

If you code lyrics attachment manipulations the way described in section 3.2, these manipulations apply *to all lyrics lines* of the referred staff *simultaneously*. In order to address each lyrics line separately, use the command

```
\verses{⟨man1⟩,⟨man2⟩,...}
```

where $\langle\text{man1}\rangle$ is a sequence of commands to manipulate the upper lyrics line, $\langle\text{man2}\rangle$ to manipulate the second lyrics line from above, and so on. These command sequences may be empty. Delimiting commas after the last non-empty list entry can be omitted. For example,

```
\verses{,,\beginmel,\golyr{abc}}
```

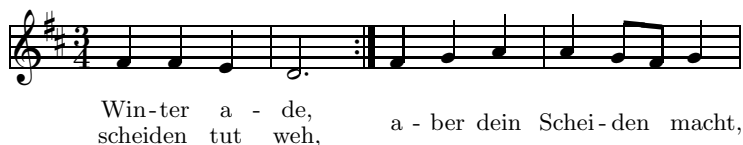
leaves the upper two lyrics lines unaltered, initiates a melisma at the third, performs a jump to label `abc` at the fourth, and leaves all further ones (whatever their number may be) unaltered. As another example, the font change for the german lyrics of the above excerpt is accomplished by

```
\znotes\verses{,\lyrlayout{\footnotesize\it}}\en
```

before the first note is coded.

3.6.2 Vertical positioning

A block of multiple lyrics lines is vertically centered around the position which a single lyrics line would have taken; thus you can do things like



without having to adjust the vertical position within the piece. (See the source of this manual for the coding of this example.)

The vertical distance between the baselines of multiple verses is appropriate for the default 10 points font size of plain \TeX . However, the size-switching commands `\smalltype`, `\medtype` etc. do not implicitly adjust the vertical distance to the chosen font size. Since the vertical distance of lyrics is controlled by the concept of a *lyrics strut*⁷, you should perform this adjustment explicitly with the command

```
\setlyrstrut
```

directly after each “global” font size change (i.e. after each size change not issued within `\lyrlayout{...}`; see 3.5.2) unless you do not want the spacing to be adjusted. (\TeX erts only: If you like to define your own vertical spacing, you may redefine the `\lyrstrutbox` yourself.)

3.7 Auxiliary lyrics

If you have typeset two voices of a polyphonic arrangement within one staff, you normally want to give each voice its own lyrics lines. Usually, the two voices are distinguished by means of the notes’ stem direction; the lyrics of the up-stemmed voice are typeset above, those of the down-stemmed voice below the staff. `musixlyr` provides the facility to maintain such two lyrics “streams” per staff; one of them — which is handled exactly as described so far — will be called *main lyrics*, the second one *auxiliary lyrics*.

⁷In analogy to \TeX ’s `\strut`; see *The \TeX book* by D. E. KNUTH.

3.7.1 Addressing auxiliary lyrics

Only two rules are to be obeyed for directing commands to auxiliary lyrics:

1. Enclose everything concerning auxiliary lyrics in:

```
\auxlyr{...}
```

For example, assign lyrics line `verse1` to staff 2 as auxiliary lyrics by:

```
\auxlyr{\assignlyrics{2}{verse1}}
```

For example, begin a melisma at the next note of the “auxiliary” voice by:

```
\auxlyr{\beginmel}
```

(For $\text{T}_{\text{E}}\text{X}$ perts: If the argument of `\auxlyr` consists of only one token, the braces can be omitted, so that `\auxlyr\beginmel` does the same.)

2. There are exactly two exceptions to the first rule. The first one are the `\setlyrics`, `\appendlyrics`, and `\copylyrics` commands which need not be enclosed in `\auxlyr{...}`; their function is independent of the later use of the defined lyrics lines as either main or auxiliary lyrics. The second exception concerns lyrics commands issued as type-1 $\text{T}_{\text{E}}\text{X}$ strings in a **PMX** source; see below for details.

Auxiliary lyrics are lifted above the staff by default, i. e. they have the same vertical position as main lyrics belonging to the staff above the current staff. If this causes collisions with the main lyrics of the upper staff, you have to shift the lyrics away from each other using the vertical spacing commands discussed in section 3.4.

3.7.2 Automatic attachment of auxiliary lyrics

While it is easy to attach only one lyrics “stream” to a single voice within a system automatically, the attachment of both main and auxiliary lyrics involves the task of determining which note belongs to which voice. Unfortunately, $\text{MusiX}_{\text{T}_{\text{E}}\text{X}}$ does not explicitly state a note’s membership of a particular voice; therefore, a heuristic method has to be used to gather this information. However, the situation is completely different if you prepare your scores with the **PMX**⁸ preprocessor. The following paragraphs explain managing the lyrics attachment automatism for both cases — **PMX** and non-**PMX** — separately.

Using PMX. (This paragraph requires some **PMX** knowledge.) For **PMX**-prepared scores (which `musixlyr` detects automagically⁹), the distinction between the two voices of a staff works fully automatic. This applies to both the automatic lyrics attachment as well as explicit lyrics manipulations — i. e., you do not have to use `\auxlyr{...}` for manipulating auxiliary lyrics. (This only applies to manipulations that appear in a voice’s context, i. e. within a type-1 $\text{T}_{\text{E}}\text{X}$ string.) For reference, see the following example¹⁰,

The image shows a musical staff with a treble clef and a key signature of one sharp (F#). The melody consists of several notes, some with slurs. Below the staff, the lyrics are written: "all der ir - - ren Qua - - len los." The first line of lyrics is positioned above the staff, and the second line is positioned below the staff, demonstrating how auxiliary lyrics are placed relative to the main staff.

which can be obtained by the following **PMX** code (processed with **PMX** version 2.3):

⁸If you don’t know what **PMX** is, have a look at: <http://icking-music-archive.sunsite.dk/musixtex/software/pmx/>

⁹This automatic **PMX** detection may cause problems for processing **PMX** scores prepared with the **PMX**-unaware `musixlyr` version 1.0. See section 3.10.5 about how to solve this backward-incompatibility issue.

¹⁰Excerpt from “Waldesnacht, du wunderkühle” by JOHANNES BRAHMS (*Eine kleine Melodie*, Strube, München 1989)

```

---
\input musixtex           % These commands are necessary
\input pmx                % because pmx.tex MUST be \input
\input musixlyr          % before musixlyr.tex!

\setlyrics{soprano}       % \setlyrics and \copylyrics
  {all der ir-ren Qua-len los.} % need not be enclosed in
\copylyrics{soprano}{alto} % \auxlyr{...}.

\assignlyrics1{alto}
\auxlyr{\assignlyrics1{soprano}
  \lyrraise1{a 3\Interligne}}

\let\BM\beginmel         % For brevity within the subsequent
\let\EM\endmel           % music coding recommended.
---
1 1 4 4 0 0 3 2 0 5 20 0

t
./

% 1st and 2nd bar:
r4-6 r2-6 | r4-8 d24s e4 //
c25n a4 | \BM\ f2d s+4 \isu0e{.8}\ e4 /
% 3rd bar:
\BM\ c s \EM\ d2r s d4 //
\ts0e0\EM\ e s+3 dn \BM\ d2 s /
% 4th bar:
d2 a // % Note that \BM and \EM
\EM\ d s c / % apply either to main
% 5th bar: % or to auxiliary lyrics,
a r+0 // % depending on which voice
d of rb / % they appear in.

```

By default, main lyrics are attached to the lower voice (whose lines end with // in the **PMX** source), auxiliary lyrics to the upper voice; due to **PMX** internals, this ensures that if only one voice is present, main lyrics are attached to it. However, situations exist in which it is desirable to have main lyrics attached to the upper voice, e.g. when you have only main lyrics and want to synchronize them with the soprano voice instead of the alto voice (both of them set on the same staff). In this case, you can toggle the attachment of lyrics to voices with the command

```
\lyrmodealter{<n>}
```

for staff number $\langle n \rangle$; the command

```
\lyrmodenormal{<n>}
```

restores the default behaviour. If $\langle n \rangle = 0$ then the change affects all staves.

Using “plain” MusiX_TE_X without PMX. In this context, there are two basically different modes for lyrics attachment:

1. In *normal mode* (which is active by default) only main lyrics are provided automatically as described in section 3.2.1. All auxiliary lyrics must be issued manually by saying `\auxlyr\lyr` before the respective note. This mode is useful when you do not have auxiliary lyrics at all or when auxiliary lyrics occur only occasionally within the piece.

2. In *alternative mode*, both main and auxiliary lyrics are provided automatically. The two voices of a staff are distinguished by means of their notes' stem direction; the lower voice gets main lyrics, the upper one auxiliary lyrics. This implies that the automatism fails at stemless notes¹¹, thus you have to provide lyrics manually by saying `\lyr` or `\auxlyr\lyr`, respectively, before such notes. This mode is meant for staves which continuously contain two polyphonic voices with separate lyrics.

The appropriate mode can be chosen staff-wise: The command

```
\lyrmodealter{<n>}
```

enables alternative mode for staff number `<n>`; the command

```
\lyrmodenormal{<n>}
```

restores normal mode. As stated above, `<n> = 0` performs the choice affecting all staves.

For illustration, have a look at the following example¹²:

The image shows a musical staff with two voices. The upper voice (soprano) has lyrics "bring her die Gans," and the lower voice (alto) has lyrics "bring her die Gans,". The notes are connected by slurs, and the lyrics are positioned above and below the staff respectively.

The lyrics were defined as follows:

```
\setlyrics{soprano}{bring her die Gans,}
\copylyrics{soprano}{alto}

\assignlyrics1{alto}
\auxlyr{\assignlyrics1{soprano}}

\lyrmodealter0           % provide main + aux. lyrics automatically
```

This is the complete notes section:

```
\Notes\zqu g\ql e\en
\bar
\Notes\zqu i\beginmel\ibslurdOf\iblOf{-1}\qbOf\en
\Notes\qb0e\en
\Notes\auxlyr\beginmel\ibsluru1i\zqup i\qb0d\en
\Notes\tqb0e\en
\Notes\endmel\tbslurd0e\ql c\en
\notes\ibbu0h{-1}\qb0h\en
\notes\tqh0g\en
\Notes\auxlyr\endmel\tbsluru1h\zqu h\ql c\en
\bar
\NOTEs\auxlyr\lyr\zwh i%      The lyrics of the whole notes
\lyr\wh b\en                %      must be given manually.
```

Note that, as opposed to using **PMX**, there is no special provision for attaching main lyrics to the upper voice. If you desire a single lyrics “thread” synchronized with the upper voice, please use auxiliary lyrics instead and position them below the staff (see section 3.4). Alternatively, you may use main lyrics in normal lyrics mode and code the lower voice completely non-spacing (in the sense explained in 3.2.1), the upper voice completely spacing; this should not be too complicated for mainly homophonic pieces.

¹¹Ancient notes, namely breve, longa, and maxima, are also treated as stemless notes by MusiX_{TEX}'s internals.

¹²Excerpt from “Audite nova” by ORLANDO DI LASSO (*ars musica* vol. 4, Möseler Verlag Wolfenbüttel/Zürich)

3.8 Lyrics at multi-staff instruments

So far in this manual, the term *staff* has always been used for what MusiX_{TEX} precisely calls an *instrument*. This section, however, correctly adopts the MusiX_{TEX} terminology.

Lyrics can be assigned to staves of a multi-staff instrument. To do so, use

```
\assignlyricsmulti{<i>}{<s>}{...}
```

instead of `\assignlyrics...`; here, `<s>` addresses the desired staff number (counted from bottom to top) of instrument `<i>`.

Nevertheless, `\assignlyrics{<i>}{...}` can be used as well if instrument `<i>` has multiple staves; in this case, it is equivalent to `\assignlyricsmulti{<i>}{1}{...}`, i.e. it affects only the first staff (and not all staves, as may be presumed).

In the same spirit, the following “multi” commands replace the standard ones within a multi-staff instrument context:

```
\lyrraisemulti
\lyrmodenormalmulti
\lyrmodealtermulti
```

All other `musixlyr` commands or features can be used the same way as within a single-staff instrument context.

Using lyrics at multi-staff instruments can be useful, for example, for typesetting songs with piano accompaniment:¹³

The image shows a musical score for piano and voice. The piano part is in the left hand, and the voice part is in the right hand. The lyrics are: Gu-ter Mond, du gehst so — stil - le.

The lyrics of this example are defined as follows:

```
\setlyrics{verse}{Gu-ter Mond, du gehst_ so_ stil_-le}
\assignlyricsmulti{1}{2}{verse}
\lyrraisemulti {1}{2}{\internote}
```

3.9 Managing lyrics assignment according to current context

Managing the properties of lyrics lines assignment to instruments by absolute instrument numbers — as it is done with the `\assignlyrics`, `\lyrraise`, and `\lyrmodealter` commands — may cause difficulties whenever the score’s number and order of instruments shall be rearranged; in this case instrument numbers change, thus the instrument number parameters of the mentioned lyrics-related commands have to be adjusted.

This is especially unpleasant where such a rearrangement is done automatically, e.g. by **PMX**’s part extraction facility `scor2prt`; this is intended to work without any manual editing of the generated part files.

To address this issue, `musixlyr` provides *context-dependent* variants of the commands mentioned above. For example: Instead of `\assignlyrics{2}{soprano}`, you may say something like:

```
\notes...&\assignlyricshere{soprano}...\en
```

¹³Example taken from songbook *Am Brunnen vor dem Tore*, Delphin Verlag München/Zürich 1981

`\assignlyricshere{⟨name(s)⟩}` assigns the given verse `⟨name(s)⟩` to the instrument/staff whose context it occurs in within the `\notes... \en` construct; no absolute instrument/staff number needs to be specified.

For example, the following two code fragments are equivalent:

<pre>\assignlyrics {1} {acct}% \assignlyricsmulti{2}{2}{verse}% \Notes\qa c&\qa e \qa g\en</pre>	<pre>\Notes\assignlyricshere{acct}\qa c% &\qa e% \assignlyricshere{verse}\qa g\en</pre>
--	--

Using **PMX**, the context-dependent commands must be given as type-1 inline T_EX strings. Here is the same example, coded as **PMX**input:

<pre>\assignlyrics {1} {acct}\ \assignlyricsmulti{2}{2}{verse}\ c44 / e44 / g44 /</pre>	<pre>\assignlyricshere{acct}\ c44 / e44 / \assignlyricshere{verse}\ g44 /</pre>
---	--

The first one (left) results in wrong lyrics assignments in *scor2prt* output; the second one (right) is robust against part extraction.

This is the complete set of “here” commands that can be applied the same way:

```
\assignlyricshere
\lyrraisehere
\lyrmodenormalhere
\lyrmodealterhere
```

All other `musixlyr` commands are context-driven anyway, i. e. using the mentioned “here” commands you can completely omit any absolute instrument/staff number assignment.

3.10 Miscellaneous

3.10.1 Avoiding problems after `\endpiece... \startpiece`

Sometimes, after having suspended the music with `\endpiece` and restarted it with `\startpiece` (or the respective `\...extract` commands), the lyrics lines are totally misaligned because the internal lyrics positioning parameters are not reset properly. Saying

```
\resetlyrics
```

before the beginning of the new music section should solve this problem.

3.10.2 Improved centering of words with punctuation marks

Centered words with an attached punctuation mark often seem a bit left-shifted since the punctuation mark also counts for the centering. To have a word centered solely with respect to its letters, enclose the punctuation mark in `\rlap{...}` as it is done in the example of section 3.6.2. Unfortunately, this circumvents the lyrics shifting automatism, and is incompatible with the use of `\lyrnop`.

3.10.3 Verse numbers

There is no special command for this topic. See the following example:



1. What shall we do with a drun - ken sail - or?
2. Take him and shake him and try to a - wake him,
3. Give him a dose of salt and wa - ter,
4. Give him a dash with a bo - som's rub - ber,

To see how this is accomplished, look for the example’s coding within `mxlyrdoc.tex`.

Section 3.2.6 explains an alternative method for setting vertically aligned verse numbers.

3.10.4 “Lyrics slurs”

If two consecutive syllables of a lyrics line share the same note (as in the second verse of the above example), you may want to link them together with a small slur under them like [this](#). To do this, use the command `\lyrlink` instead of the respective blank space within the `\setlyrics` definition.

If any of the adjacent characters extend under their \TeX baseline (like ‘g’, ‘j’, ‘p’, etc.), use `\lowlyrlink` which lowers the slur accordingly (e. g.: sing [gladly](#)).

3.10.5 Restoring the PMX-unaware behaviour of musixlyr 1.0

As stated in section 3.7.2, `musixlyr` distinguishes between main and auxiliary lyrics context in a special way when it detects that the score has been set up using **PMX**. However, this automatic **PMX** detection may cause problems for processing **PMX** scores prepared with the **PMX**-unaware `musixlyr` version 1.0. To solve this, you can force the old `musixlyr` behaviour by an explicit macro redefinition as follows:

Within `musixlyr.tex`, find the code fragment

```
\ifx\nextvoice\undefined
  \def\decide@lyrmode{%
    \call@lyrmode
    ...
    \fi\fi\fi\fi\fi\fi}
\else
```

(which is about 25 lines long). Copy it, excluding the first and the last line, to your score source and surround it by the `\makeatletter... \makeatother` command pair. This should then look like this:

```
\makeatletter
  \def\decide@lyrmode{%
    \call@lyrmode
    ...
    \fi\fi\fi\fi\fi\fi}
\makeatother
```

To restore the **PMX**-aware `musixlyr` behaviour of version 1.1 and newer, copy the *second* `\decide@lyrmode` definition from `musixlyr.tex` to the desired place in your score and surround it with `\makeatletter... \makeatother`.

To do these redefinitions within a **PMX** input file, it is probably always best to wrap them in macro definitions within the **PMX** file preamble. This way, you can switch between the “old” and “new” `musixlyr` behaviour by simply calling these macros in **PMX**’s inline- \TeX style.

4 Command index

<code>\appendlyrics</code> , 5, 7, 15	<code>\lyrlink</code> , 20
<code>\assignlyrics</code> , 5, 6, 13, 17–19	<code>\lyrmodealter</code> , 16–18
<code>\assignlyricshere</code> , 18, 19	<code>\lyrmodealterhere</code> , 19
<code>\assignlyricsmulti</code> , 18, 19	<code>\lyrmodealtermulti</code> , 18
<code>\auxlyr</code> , 2, 15–17	<code>\lyrmodenormal</code> , 16, 17
<code>\auxsetsongraise</code> , 12	<code>\lyrmodenormalhere</code> , 19
<code>\beginmel</code> , 6, 7, 9	<code>\lyrmodenormalmulti</code> , 18
<code>\copylyrics</code> , 2, 5, 6, 9, 10, 15	<code>\lyrnop</code> , 8, 9, 19
<code>\enableauxlyrics</code> , 2	<code>\lyroffset</code> , 3, 10
<code>\endmel</code> , 6, 7, 9	<code>\lyrpt</code> , 8, 9
<code>\forcelyrhyphensfalse</code> , 12	<code>\lyrraise</code> , 12, 18
<code>\forcelyrhyphenstrue</code> , 12	<code>\lyrraisehere</code> , 19
<code>\golyr</code> , 8, 9	<code>\lyrraisemulti</code> , 18
<code>\lclyr</code> , 9	<code>\lyrrule</code> , 10
<code>\leftlyrfalse</code> , 9	<code>\lyrruleend</code> , 10
<code>\leftlyrtrue</code> , 9	<code>\lyrstrutbox</code> , 14
<code>\llabel</code> , 8, 9	<code>\minlyrrulelength</code> , 12
<code>\llyr</code> , 9	<code>\minlyrspace</code> , 11, 12
<code>\lowlyrlink</code> , 20	<code>\minmulthyphens</code> , 13
<code>\lyr</code> , 9, 16, 17	<code>\nolyr</code> , 5, 8, 9
<code>\lyrhyphenchar</code> , 4, 13	<code>\oldlyrlinestart</code> , 3
<code>\lyric</code> , 10, 11	<code>\resetlyrics</code> , 19
<code>\lyric*</code> , 10, 11	<code>\setlyrics</code> , 2–10, 13, 15
<code>\lyrich</code> , 10, 11	<code>\setlyrstrut</code> , 14
<code>\lyrich*</code> , 10, 11	<code>\setsongraise</code> , 12
<code>\lyricsoff</code> , 8	<code>\showlyrshifttrue</code> , 12
<code>\lyricson</code> , 8	<code>\verses</code> , 7, 10, 11, 14
<code>\lyrlayout</code> , 13	

5 Software license

`musixlyr` (including this manual) is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version.

`musixlyr` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with `musixlyr`; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Author: Rainer Dunker
 Wachtelweg 31
 85 591 Vaterstetten
 Germany
 E-mail: rainer.dunker@web.de