
OpenLyrics Documentation

Release 0.7

Martin Zibricky, Raoul Snyman

March 24, 2010

Contents

1	Introduction	1
1.1	About	1
1.2	History	1
1.3	Release Numbering	1
2	OpenLyrics concepts	3
3	Data Format	5
3.1	Basic Structure	6
3.2	Features	7
3.3	Required Data Items	7
3.4	Metadata	8
3.5	Encoding and Filenames	9
3.6	Song Properties	9
3.7	Song lyrics	14
3.8	Advanced Example	18
4	Validation	21
4.1	Prerequisites	21
4.2	Using bundled tool	21
4.3	Other tools	21
4.4	RelaxNG xml schema	21
5	Conversion from other formats	23
5.1	OpenSong	23
5.2	Other formats	23
6	List of Chords	25
7	List of Themes	27
8	OpenLyrics FAQ	29
8.1	How do I...	29
9	Glossary	31
10	Changes in OpenLyrics	33
10.1	Release 0.7 (19 Mar 2010)	33
10.2	Release 0.6 (22 Dec 2009)	33

10.3	Release 0.5 (06 Dec 2009) - final draft	33
10.4	Release 0.4 (21 Nov 2009) - draft	34
10.5	Release 0.3 (18 Nov 2009) - draft	34
10.6	Release 0.2 (16 Nov 2009) - draft	34
10.7	Release 0.1 (28 Nov 2008) - draft	34
11	Projects using OpenLyrics	35
	Index	37

Introduction

1.1 About

OpenLyrics is a free, open XML standard for Christian worship songs. The goal of OpenLyrics is to provide an application-independent and operating system-independent song format for interoperability between applications.

1.2 History

The first version of OpenLyrics was created in 2008, when the OpenLP project leader approached the ChangingSong project leader, and proposed cooperation between OpenLP and ChangingSong to improve data exchange between the two projects. They agreed that a good first step would be to create an independent interoperable data format to provide better song exchange between the two applications.

Furthermore, experiences the leader of the ChangingSong had with the OpenSong project's XML format for songs made him aware that the OpenSong format was not sufficient for the proposed features in ChangingSong, and thus a new format would be necessary in order to implement many of the more advanced features the project planned to develop.

Upon planning the OpenLyrics format, the two leaders decided to make the format as open and inclusive as possible so that other presentation projects, both open source and proprietary, could use this new format as well.

To that end, they decided to make OpenLyrics an XML format. XML is a well established standard with solid support in many programming languages, and there are a plethora of XML libraries for manipulating XML.

The current design of OpenLyrics is based on the OpenSong data format along with some features suggested by users, particularly the ability to use [multiple languages for a song \(forum\)](#).

1.3 Release Numbering

OpenLyrics uses the following release numbering scheme:

`X.X_pX`

where `X.X` is the major release number, indicating the data format version and `pX` is the optional minor release number, used when incidental files (documentation, examples, and the like) are updated.

Therefore, this would be a valid release number, indicating a new data format:

`0.6`

And this would be the minor release made when we correct typos in documentation, add an example song, etc:

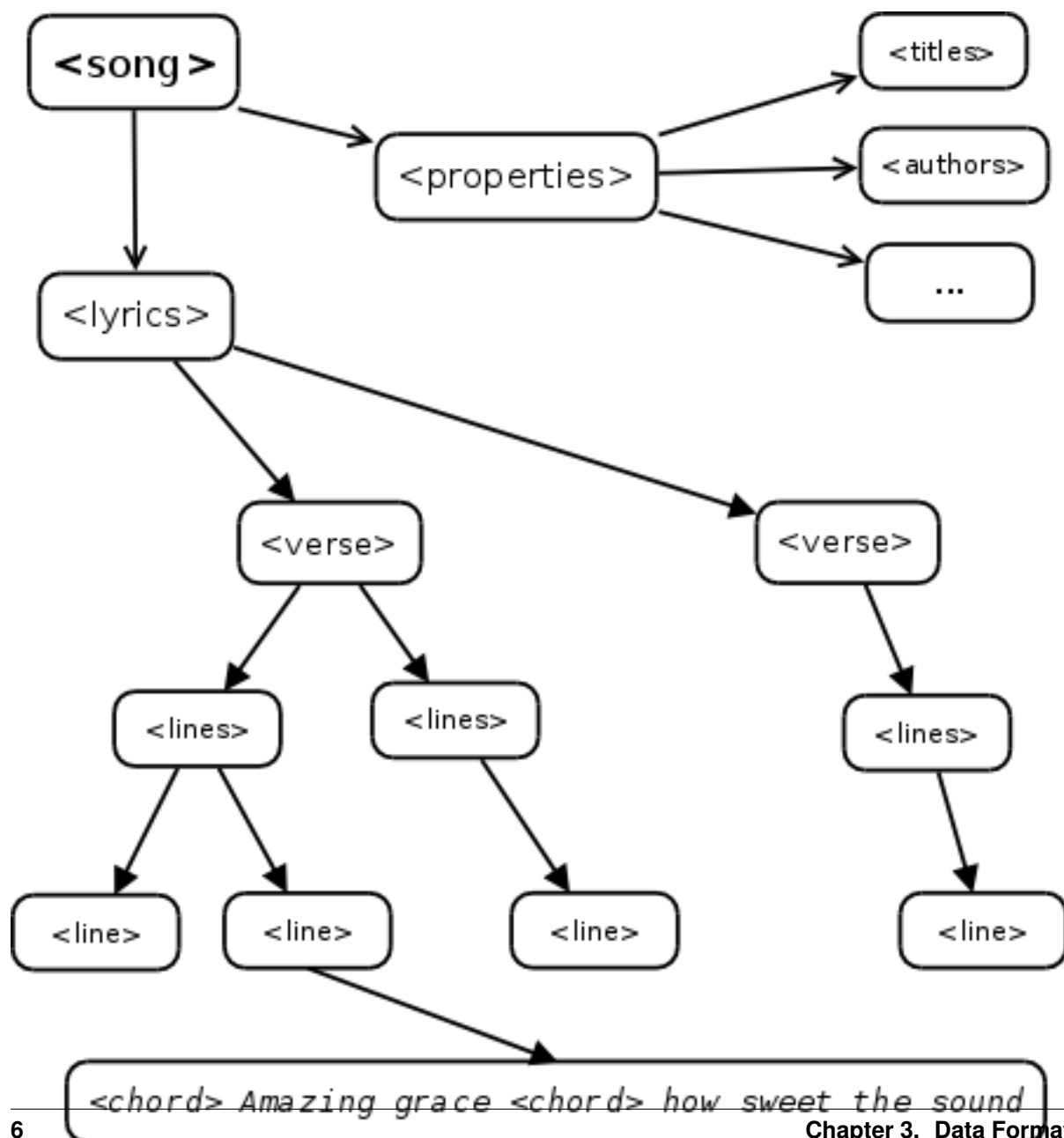
0.6_p1

OpenLyrics concepts

needs to be written

Data Format

3.1 Basic Structure



3.2 Features

categories <theme>

CCLI support <ccliNo>

chords <chord name="D">

comments in lyrics <verse><lines><comment/></lines></verse>

date of song release <releaseDate>

format version <song version="0.6>

keywords for searching <keywords>

last modification time <song modifiedDate="">

lines of text <line>

multiple authors <authors>

multiple categories <themes>

multiple song titles <titles>

multiple user-defined items <comments>

music properties <transposition> <tempo> <key>

namespace <song xmlns="http://openlyrics.info/namespace/2009/song">

parts <lines part="men">

slides <verse>

song book <collection> <trackNo>

song metadata <song version=""> <song createdIn=""> <song modifiedIn=""> <song modifiedDate="">

song translator <author type="translator" lang="cs">

song variant <variant>

song version <customVersion>

tagging verse type <verse name="v1">

translated lyrics <verse name="v1" lang="en">

translated song title <title lang="en">

translated theme <theme lang="en">

transliterated lyrics <verse name="v1" lang="en" translit="he">

transliterated song title <title lang="en" translit="he">

transliterated theme <theme lang="en" translit="he">

transposition <transposition>

user-defined item <comment>

verse order <verseOrder>

3.3 Required Data Items

The song, containing only necessary data items, follows:

```
<song xmlns="http://openlyrics.info/namespace/2009/song"
      version="0.6"
      createdIn="OpenLP 1.9.0"
      modifiedIn="ChangingSong 0.0.1"
      modifiedDate="2010-01-28T13:15:30+01:00">
  <properties>
    <titles>
      <title>Amazing Grace</title>
    </titles>
  </properties>
  <lyrics>
    <verse name="v1">
      <lines>
        <line>Amazing grace how sweet the sound</line>
      </lines>
    </verse>
  </lyrics>
</song>
```

As you can see from the previous example, a minimalistic song should contain only:

- metadata
- title
- verse with one line of text

Elements with empty values aren't allowed. If a data item is not present in the song, the tag, where the data would be put, should not be in xml.

3.4 Metadata

Metadata are **required** to be present in every song. They should ease debugging of applications using OpenLyrics.

Metadata are enclosed in tag `<song>` as its attributes:

```
<song xmlns="http://openlyrics.info/namespace/2009/song"
      version="0.6"
      createdIn="OpenLP 1.9.0"
      modifiedIn="ChangingSong 0.0.1"
      modifiedDate="2010-01-28T13:15:30+01:00">
```

xmlns Defines a xml namespace. The value should be always `http://openlyrics.info/namespace/2009/song`

version Version of the OpenLyrics format used by a song. This allows applications to notify users, if the application doesn't support newer versions of OpenLyrics.

createdIn String to identify the application where a song was created for the first time. This attribute should be set when a new song is created. It should not be changed with additional updates and modification to the song. Even when the song is edited in another application. Recommended content of this attribute is *application name* and *version* like `OpenLP 1.9.0`.

modifiedIn String to identify the application where a song was edited for the last time. This attribute should be set with every modification. Recommended content of this attribute is *application name* and *version* like `OpenLP 1.9.0`.

modifiedDate Date and time of last modification. This attribute should be set with every modification. The used format of date is [ISO 8601](#). It should be in the format `YYYY-MM-DDThh:mm:ss±[hh]:[mm]`.

3.5 Encoding and Filenames

3.5.1 Encoding

I recommend using [UTF-8](#) encoding for the content of xml files in OpenLyrics format. *UTF-8* is well supported among programming libraries.

3.5.2 Filenames

In regards to filenames, the recommendation is to use such a name which will well identify the song just by looking at the filename. For the file could be used a combination of fields `<titles>`, `<variant>` and/or `<authors>`. Since OpenLyrics is a xml based format, filenames should contain the extension `.xml`

Examples:

```
Amazing Grace.xml
Amazing Grace (old hymn).xml
Amazing Grace (John Newton).xml
```

It would be nice, if songs containing non ASCII characters in its title, use also nos ASCII characters in filenames. These days all major operating systems should support localized characters in filenames. However, there are some limitation in this approach. Not all archive formats handle localized filenames well. For example, one of most used archive formats, [ZIP](#). On the other hand, the format [7-Zip](#) handles it well.

3.6 Song Properties

Description of all possible elements enclosed in tag `<properties>`. Elements enclosed in this tag may be any arbitrary order. For example it doesn't matter if tag `<titles>` occurs before `<authors>`:

```
<titles><title>Amazing Grace</title></titles>
<authors><author>John Newton</author></authors>
```

Or `<titles>` occurs after `<authors>`:

```
<authors><author>John Newton</author></authors>
<titles><title>Amazing Grace</title></titles>
```

An application implementing OpenLyrics should not depend on any order of elements enclosed in the tag “`<properties>`”.

3.6.1 Titles

Title is a **mandatory element**. Every song must contain at least one title:

```
<titles><title>Amazing Grace</title></titles>
```

There could be more titles:

```
<titles>
  <title>Amazing Grace</title>
  <title>Amazing</title>
</titles>
```

You can define attribute `lang=""`. It says what is the language of the title. The value of this attribute should be in the format `xx` or `xx-YY` where `xx` is an [ISO-639 language code](#) and `YY` is an [country code](#). For more details see [bcp47](#).

It comes handy when the song is translated from one language to another and there is a need to know the titles in other languages or the song contains lyrics in multiple languages:

```
<titles>
  <title lang="en">Amazing Grace</title>
  <title lang="de">Staunenswerte Gnade</title>
  <title lang="pl">Cudowna Boża łaska</title>
</titles>
```

Additionally, it is possible use attribute `original="true"`. This attribute expresses that a title is the title of the original song:

```
<titles>
  <title lang="en" original="true">Amazing Grace</title>
  <title lang="pl">Cudowna Boża łaska</title>
</titles>
```

3.6.2 Authors

Authors is an optional element. When this element is present in the song, there should be at least one subelement `<author>`:

```
<authors><author>John Newton</author></authors>
```

There could be more authors:

```
<authors>
  <author>John Newton</author>
  <author>Johannes Newton</author>
</authors>
```

Three types of authors can be distinguished:

- *author of words*:

```
<author type="words">John Newton</author>
```

- *author of music*:

```
<author type="music">John Newton</author>
```

- *translator*:

```
<author type="translation" lang="cs">Jan Ľútň</author>
```

The translator type must in addition contain the attribute `lang=""`. Value of this attribute should be in the format `xx` or `xx-YY`. `xx` means language code and `YY` means country code.

3.6.3 Copyright

This element should contain copyright information. In some countries it is necessary to display copyright information during presentation. In this situation this comes handy.

This element could look like:

```
<copyright>public domain</copyright>
```

Or for example:

```
<copyright>1998 Vineyard Songs</copyright>
```

3.6.4 CCLI Number

CCLI stands for *Christian Copyright Licensing International*. CCLI offers copyright licensing of songs and other resource materials for use in Christian worship. For registered churches CCLI offers songs and other resources for download. At CCLI an ID is assigned to every song. This element provides integration with CCLI.

CCLI number (id) must be a positive integer:

```
<ccliNo>22025</ccliNo>
```

3.6.5 Release Date

Release date is a date/time when the song was released or published.

It can be just year:

```
<releaseDate>1779</releaseDate>
```

year-month:

```
<releaseDate>1779-09</releaseDate>
```

year-month-day:

```
<releaseDate>1779-12-30</releaseDate>
```

year-month-day and time:

```
<releaseDate>1779-12-31T13:15</releaseDate>
```

3.6.6 Transposition

This element is used when there is a need to move the key or the pitch of chords up or down. The value must be a positive or negative integer.

The negative integer moves the pitch down by a fixed number of semitones:

```
<transposition>-3</transposition>
```

The positive integer moves the pitch up by a fixed number of semitones:

```
<transposition>4</transposition>
```

3.6.7 Tempo

Tempo means how the song should be played. It could be expressed in beats per minute (bpm) or as any text value. Tempo in bpm must be a positive integer in the range 30-250:

```
<tempo type="bpm">90</tempo>
```

Tempo expressed as text can contain any arbitrary text. For example `Very Fast`, `Fast`, `Moderate`, `Slow`, `Very Slow`, etc.:

```
<tempo type="text">Moderate</tempo>
```

3.6.8 Key

Key determines the musical scale of a song. The value could be `A`, `B`, `C#`, `D`, `Eb`, `F#`, `Ab`, etc.

Example:

```
<key>Eb</key>
```

3.6.9 Variant

Variant should be used in situation, where there are 2 songs with the same title and it is needed to distinguish those songs.

For example there could be two songs with the title *Amazing grace*. One song published many years ago and one song published by a well known band, say for instance *Newsboys*.

For the old song it could be:

```
<variant>old hymn</variant>
```

For the song from a well known band:

```
<variant>Newsboys</variant>
```

3.6.10 Publisher

Name of the publisher of the song:

```
<publisher>Sparrow Records</publisher>
```

3.6.11 Custom Version

Many songs aren't written at once. When a new song is written, it usually contains just a title and lyrics. In the future the song will be updated and more data will be added. This could help users distinguish different versions of the same song.

This element can contain any arbitrary text which could help the user to distinguish various song versions.

It could contain for example a version number:

```
<customVersion>0.99</customVersion>
```

or date:


```
<customVersion>2010-02-04</customVersion>
```

or anything else:

```
<customVersion>this is previous version</customVersion>
```

3.6.12 Keywords

Keywords are used to get more precise results when searching for a song in the song database.

For the song *Amazing Grace* it could be:

```
<keywords>amazing grace, how sweet the sound, a wretch like me</keywords>
```

3.6.13 Verse Order

Determines the order of verses in the lyrics. In lyrics part every verse is enclosed in tag `<verse>`. Every verse should have a different one word name.

Every word in `<verseOrder>` refers to name of verse in the `<lyrics>`. Verse name can appear in `<verseOrder>` multiple times.

Verse names should be in lowercase.

For example when in the lyrics part are verses with names *v1*, *v2*, and *c*, this element could be:

```
<verseOrder>v1 c v2 c v1 c</verseOrder>
```

3.6.14 Songbooks

If a song comes from any collection or songbook, here should be noted the name of the songbook/collection and number of a song in that songbook. For songbook name is used attribute `name=""` and for number attribute `entry=""`. Both attributes can contain any text:

```
<songbooks>
  <songbook name="Name of a songbook or collection" entry="48"/>
</songbooks>
```

The attribute `name=""` is mandatory but `entry=""` is not:

```
<songbooks>
  <songbook name="Name of a songbook or collection"/>
</songbooks>
```

The attribute `name=""` is mandatory but `entry=""` is not:

```
<songbooks>
  <songbook name="This is a Songbook Name" entry="48"/>
  <songbook name="Name of a songbook without number"/>
  <songbook name="Name of a songbook" entry="84c"/>
</songbooks>
```

3.6.15 Themes

Themes are used to categorize songs. Having songs categorized can be useful when choosing songs for a ceremony or for a particular topic.

There can be one or more themes:

```
<themes><theme>Adoration</theme></themes>
```

As additional attributes could be an `id=""` and/or `lang=""`:

```
<themes>
  <theme>Adoration</theme>
  <theme id="1" lang="en-US">Grace</theme>
  <theme id="2" lang="en-US">Praise</theme>
  <theme id="3" lang="en-US">Salvation</theme>
  <theme id="1" lang="pt-BR">Graça</theme>
  <theme id="2" lang="pt-BR">Adoração</theme>
  <theme id="3" lang="pt-BR">Salvação</theme>
</themes>
```

The `id` attribute should be used when using a theme from a [standardized CCLI list](#). The list can be found in the downloadable OpenLyrics archive in file with name `themelist.txt`. The value of `id` is the line number of a particular theme in this file. Standardized themes with `id` should ease assigning translated themes to songs in an application.

`lang` defines a language of a theme. Value of this attribute should be in the format `xx` or `xx-YY`. `xx` means language code and `YY` means country code.

3.6.16 Comments

This field is for other additional unspecified user data. There can be more items. The value can be any text:

```
<comments>
  <comment>One of the most popular songs in our congregation.</comment>
  <comment>We sing this song often.</comment>
</comments>
```

3.7 Song lyrics

Description of the possible syntax enclosed in tag `<lyrics>`. This tag contain text of a song and other stuff related to it.

Lyrics part of OpenLyrics format will mostly contain elements like `<verse>`, `<lines>` or `<line>`. A song should contain at least **one verse with one line**.

```
<lyrics>
  <verse name="v1">
    <lines>
      <line>This is the first line of the text.</line>
    </lines>
  </verse>
</lyrics>
```

In `<line>` should be enclosed all text for one line. An empty line is expressed by `<line/>` or `<line></line>`. More lines of text can be in element `<lines>`:

```
<lines>
  <line>This is the first line of the text.</line>
  <line>This is the second line of the text.</line>
</lines>
```

There can be more group of lines:

```
<verse name="v1">
  <lines>
    <line>This is the first line of the text.</line>
  </lines>
  <lines>
    <line>This is the second line of the text.</line>
  </lines>
</verse>
```

And of course it can contain more verses:

```
<lyrics>
  <verse name="v1">
    <lines><line>First line of first verse.</line></lines>
  </verse>
  <verse name="v2">
    <lines><line>First line of second verse.</line></lines>
  </verse>
</lyrics>
```

The tag `<verse>` is meant for all song parts, not only verses but also for *chorus*, *bridge*, *pre-chorus*, etc.

It is recommended to enclose in one element `<verse>` text for one slide.

3.7.1 Verse Name

For every verse an attribute `name` with a unique value is required. *There shouldn't be two or more verses the same name.* (The exception is the situation when there are present more translations of a verse.)

The recommendation for standardized verse names follows:

Name	Description
v1, v2, v3, ...	first verse, second verse, third verse, ...
c	chorus
c1, c2, ...	more choruses
p	pre-chorus
p1, p2, ...	more pre-choruses
b	bridge
b1, b2, ...	more bridges
e	ending
e1, e2, ...	more endings
v1a, v1b, v1c, ...	this schema is for splitting verse to more parts (e.g. for splitting verse over more slides)
ca, cb, ...	splitting chorus to more parts
c1a, c1b, ...	splitting chorus to more parts
pa, pb, ...	splitting pre-chorus to more parts
p1a, p1b, ...	splitting pre-chorus to more parts
ba, bb, bc, ...	splitting bridge to more parts
b1a, b1b, b1c, ...	splitting bridge to more parts
ea, eb, ec, ...	splitting ending to more parts
e1a, e1b, e1c, ...	splitting ending to more parts

In recommended naming schema are names in **lowercase**. The value of the `name` attribute could be any *one word*.

Example of a song containing two verses (*v1*, *v2*), a chorus (*c*), bridge (*b*) and ending (*e*). Second verse is splitted into more slides:

```
<lyrics>
  <verse name="v1">
    ...
  </verse>
  <verse name="v2a">
    ...
  </verse>
  <verse name="v2b">
    ...
  </verse>
  <verse name="c">
    ...
  </verse>
  <verse name="b">
    ...
  </verse>
  <verse name="e">
    ...
  </verse>
</lyrics>
```

3.7.2 Chords

OpenLyrics format allows storing chords. Having chords can be handy in some situations. For example when printing leadsheets or when presetting a song during band training.

The element containing a chord name looks like:

```
<chord name="D7"/>
```

Elements `<chord>` are mixed with the text of a song. This element should be placed immediately before the letters where it should be played:

```
<lyrics>
  <verse name="v1">
    <lines>
      <line><chord name="D7"/>Amazing grace how
        <chord name="E"/>sweet the sound</line>
      <line>That saved <chord name="A"/>a wretch
        <chord name="F#"/>like me.</line>
    </lines>
  </verse>
</lyrics>
```

At the moment there is no a fixed notation for chords. But if you would like to see some examples how chords can be written, see *chord examples*.

3.7.3 Multiple Languages (Lyrics Translations)

Lyrics translation can be useful for example in situation when the worship band is singing a song in a foreing language and wants to display translation of the text for others to understand.

Translations are at the verse level. They can be added by translating a text of a `<verse>` and by adding attribute `lang=""` to `<verse>`. The value of this attribute should be in the format `xx` or `xx-YY` where `xx` is an [ISO-639 language code](#) and `YY` is an [country code](#). For more details see [bcp47](#).

More translations of a verse should have the same value of the attribute `name=""` but different values of `lang=""`.

Example of a song containg English and German translation for the first verse:

```
<lyrics>
  <verse name="v1" lang="en">
    <lines><line>This text is in English.</line></lines>
  </verse>
  <verse name="v1" lang="de">
    <lines><line>Dieses Text ist auf Deutsch.</line></lines>
  </verse>
</lyrics>
```

Because translations are defined at the verse level, there can be also situation, that some verses have translations and some do not.

3.7.4 Transliteration

Transliteration comes handy in situation when singing for instance a Hebrew song but the congregation is not able read Hebrew aplhabet.

Transliteration allows to distinguish in one song:

- text written in original alphabet (e.g. Hebrew)
- pronunciation of original aplhabet mapped to requested alphabet (e.g. Hebrew pronunciation written in English)
- translation of the song to requested language (e.g. English translation)

Transliteration can be defined by adding attribute `translit=""` to `<title>`, `<theme>` or `<verse>`. The value of this attribute should be in the format `xx` or `xx-YY` where `xx` is an [ISO-639 language code](#) and `YY` is an [country code](#). For more details see [bcp47](#).

The attribute `translit=""` must be used in conjunction with attribute `lang=""`. This is because one writting system can be transliterated to different languages differently. For example Hebrew is transliterated differently to English and French.

In the following example the attribute `lang=""` means the language of original alphabet (Hebrew) and `translit=""` means the language to what the song was transliterated (English):

```
<verse name="v1" lang="he" translit="en">
...
</verse>
```

Example of lyrics containing original text written in Hebrew `<verse name="v1" lang="he">`, transliterated to English `<verse name="v1" lang="he" translit="en">` and translated to English `<verse name="v1" lang="en">`:

```
<lyrics>
  <verse name="v1" lang="he">
    <lines><line>הבה נגילה</line></lines>
  </verse>
  <verse name="v1" lang="he" translit="en">
    <lines><line>Hava nagila</line></lines>
  </verse>
  <verse name="v1" lang="en">
    <lines><line>Let's rejoice</line></lines>
  </verse>
</lyrics>
```

3.7.5 Verse Parts (Groups of Lines)

Using the term *verse parts* or *group of lines* could be misleading but It was hard to find any other meaningful name for this feature.

This feature means the ability marking words for different groups of people. For example some words should be sung by men and some by women.

This feature is the reason why there is the tag `<lines>`. This ability is implemented by adding attribute `part=""` to the element `<lines>`. The value of this attribute is any arbitrary text.

Example of lyrics containing one verse with some words for men and soe word for women:

```
<lyrics>
  <verse name="v1">
    <lines part="men">
      <line>First line of words sung by men.</line>
      <line>Second line of words sung by men.</line>
    </lines>
    <lines part="women">
      <line>First line of words sung by women.</line>
      <line>Second line of words sung by women.</line>
    </lines>
  </verse>
</lyrics>
```

3.7.6 Comments in Lyrics

OpenLyrics format supports adding comments. Comments must be put before a line with text. Comments could be useful for printing leadsheets. They can contain various information. For example it could contain information how to play or sing any particular text.

Example:

```
<lyrics>
  <verse name="v1">
    <lines>
      <comment>Singing loudly.</comment>
      <line>Text of verse.</line>
      <comment>Singing quietly.</comment>
      <line>Text of verse.</line>
    </lines>
  </verse>
  <verse name="c">
    <lines>
      <comment>Singing loudly.</comment>
      <line>Line content.</line>
      <line>Line content.</line>
    </lines>
  </verse>
</lyrics>
```

3.8 Advanced Example

More song examples can be found in folder “songs” distributed with the OpenLyrics ZIP archive.

Here’s an example of the XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<song xmlns="http://openlyrics.info/namespace/2009/song"
      version="0.7"
      createdIn="OpenLP 2.0"
      modifiedIn="ChangingSong 0.0.2"
      <!-- date format: ISO 8601 -->
      modifiedDate="2009-12-22T21:24:30+02:00">
  <properties>
    <titles>
      <title>Amazing Grace</title>
    </titles>
    <authors>
      <author>John Newton</author>
    </authors>
    <copyright>Public Domain</copyright>
    <ccliNo>2762836</ccliNo>
    <releaseDate>1779</releaseDate>
    <tempo type="text">moderate</tempo>
    <key>D</key>
    <verseOrder>v1 v2 v3 v4 v5 v6</verseOrder>
    <themes>
      <theme>Assurance</theme>
      <theme>Grace</theme>
      <theme>Praise</theme>
      <theme>Salvation</theme>
    </themes>
  </properties>
  <lyrics>
    <verse name="v1">
      <lines>
        <line>Amazing grace how sweet the sound</line>
        <line>That saved a wretch like me.</line>
        <line>I once was lost, but now am found,</line>
        <line>Was blind but now I see.</line>
      </lines>
    </verse>
    <verse name="v2">
      <lines>
        <line>T'was grace that taught my heart to fear,</line>
        <line>And grace my fears;</line>
        <line>How precious did that grace appear</line>
        <line>The hour I first believed.</line>
      </lines>
    </verse>
    <verse name="v3">
      <lines>
        <line>Through many dangers, toil and snares,</line>
        <line>I have already come;</line>
        <line>'Tis grace has brought me safe thus far,</line>
        <line>And grace will lead me home.</line>
      </lines>
    </verse>
    <verse name="v4">
      <lines>
        <line>When we've been there ten thousand years</line>
        <line>Bright shining as the sun,</line>
        <line>We've no less days to sing God's praise</line>
        <line>Than when we've first begun.</line>
      </lines>
    </verse>
  </lyrics>
</song>

```

Validation

4.1 Prerequisites

needs to be written

4.2 Using bundled tool

needs to be written

4.3 Other tools

needs to be written

4.4 RelaxNG xml schema

needs to be written

Conversion from other formats

5.1 OpenSong

needs to be written

5.2 Other formats

needs to be written

List of Chords

Ab Ab+ Ab4 Ab7 Ab11 Absus Absus4 Abdim Abmaj Abmaj7 Abm Abm7 A A+ A4 A6 A7 A7+ A9 A11 A13 A7sus4 A9sus Asus Asus2 Asus4 Adim Amaj Amaj7 Am A/D A/F# A/G# Am#7 Am6 Am7 Am7sus4 Am9 Am/G Amadd9 Am(add9) A# A#+ A#4 A#7 A#sus A#sus4 A#maj A#maj7 A#dim A#m A#m7 Bb Bb+ Bb4 Bb6 Bb7 Bb9 Bb11 Bbsus Bbsus4 Bbmaj Bbmaj7 Bbdim Bbm Bbm7 Bbm9 B B+ B4 B7 B7+ B7#9 B7(#9) B9 B11 B13 Bsus Bsus4 Bmaj Bmaj7 Bdim Bm B/F# BaddE B(addE) BaddE/F# Bm6 Bm7 Bmmaj7 Bm(maj7) Bmsus9 Bm(sus9) Bm7b5 C C+ C4 C6 C7 C9 C9(11) C11 Csus Csus2 Csus4 Csus9 Cmaj Cmaj7 Cm Cdim C/B Cadd2/B CaddD C(addD) Cadd9 C(add9) Cm7 Cm11 C# C#+ C#4 C#7 C#7(b5) C#sus C#sus4 C#maj C#maj7 C#dim C#m C#add9 C#(add9) C#m7 Db Db+ Db7 Dbsus Dbsus4 Dbmaj Dbmaj7 Dbdim Dbm Dbm7 D D+ D4 D6 D7 D7#9 D7(#9) D9 D11 Dsus Dsus2 Dsus4 D7sus2 D7sus4 Dmaj Dmaj7 Ddim Dm D/A D/B D/C D/C# D/E D/G D5/E Dadd9 D(add9) D9add6 D9(add6) Dm7 Dm#5 Dm(#5) Dm#7 Dm(#7) Dm/A Dm/B Dm/C Dm/C# Dm9 D# D#+ D#4 D#7 D#sus D#sus4 D#maj D#maj7 D#dim D#m D#m7 Eb Eb+ Eb4 Eb7 Ebsus Ebsus4 Ebmaj Ebmaj7 Ebdim Ebadd9 Eb(add9) Ebm Ebm7 E E+ E5 E6 E7 E7#9 E7(#9) E7b9 E7(b9) E7(11) E9 E11 Esus Esus4 Emaj Emaj7 Edim Em Em6 Em7 Em/B Em/D Em7/D Emsus4 Em(sus4) Emadd9 Em(add9) F F+ F4 F6 F7 F9 F11 Fsus Fsus4 Fmaj Fmaj7 Fdim Fm F/A F/C F/D F/G F7/A Fmaj7/A Fmaj7/C Fmaj7(#5) Fadd9 F(add9) FaddG FaddG Fm6 Fm7 Fmmaj7 F# F#+ F#7 F#9 F#11 F#sus F#sus4 F#maj F#maj7 F#dim F#m F#/E F#4 F#m6 F#m7 Gb Gb+ Gb7 Gb9 Gbsus Gbsus4 Gbmaj Gbmaj7 Gbdim Gbm Gbm7 G G+ G4 G6 G7 G7+ G7b9 G7(b9) G7#9 G7(#9) G9 G9(11) G11 Gsus Gsus4 G6sus4 G6(sus4) G7sus4 G7(sus4) Gmaj Gmaj7 Gmaj7sus4 Gmaj9 Gm Gdim Gadd9 G(add9) G/A G/B G/D G/F# Gm6 Gm7 Gm/Bb G# G#+ G#4 G#7 G#sus G#sus4 G#maj G#maj7 G#dim G#m G#m6 G#m7 G#m9maj7 G#m9(maj7)

List of Themes

needs to be written

OpenLyrics FAQ

This is a list of Frequently Asked Questions about OpenLyrics. Feel free to suggest new entries!

8.1 How do I...

needs to be written

Glossary

metadata Data not directly related to a song, but useful for debugging and other similar purposes.

song properties Additional information about a song. There can be author name, copyright information, publisher, release date, etc.

theme Clasifying songs to various groups along suitability.

Changes in OpenLyrics

10.1 Release 0.7 (19 Mar 2010)

- move attribute 'xml:lang' to just 'lang' since the element could not always contain text in another language but also transliteration
- allow empty lines by using syntax `<line/>` or `<line></line>` for that
- add 'e' (ending) as a new reserved (standardized) name for verses (including other variants like e1, e2, e1a, ea, ...): `<verse name="e">`
- add transliteration support, attribute `translit=""` can be used in elements `<title>`, `<theme>`, `<verse>`
- add multiple songbooks entries. Syntax changed from elements `<collection>` and `<trackNo>` to:

```
<songbooks>
  <songbook name="This is a Songbook Name" entry="123"/>
</songbooks>
```

- add script 'opensong2openlyrics.py' to convert OpenSong songs to OpenLyrics format
- add examples of some public domain songs or not copyrighted

10.2 Release 0.6 (22 Dec 2009)

- drop element `<customData>` (is ambiguous)

10.3 Release 0.5 (06 Dec 2009) - final draft

- Allow custom verse names `<verse name="custom_name_name">`
- custom tempo `<tempo type="custom">steadily</tempo>`
- stay with only one key (any text) `<key>C#</key>`
- allow any chord notation (any text)
- restrict ccli theme 'id' to range 1-999
- theme value can't be empty

- namespace changed from `http://www.openlyrics.info/2009/song/namespace` to `http://openlyrics.info/namespace/2009/song`
- content of an optional element is mandatory, when the element present in xml.

10.4 Release 0.4 (21 Nov 2009) - draft

- move to RelaxNG xml schema
- tag attribute change. `xml:lang` describes the language of an element. Change from `<author type="translation" xml:lang="cs">` to `<author type="translation" lang="cs">`
- xml schema: in 'author' element attribute 'lang' is required when attribute 'type' contains value 'translation'
`<author type="translation" lang="cs">`

10.5 Release 0.3 (18 Nov 2009) - draft

10.6 Release 0.2 (16 Nov 2009) - draft

10.7 Release 0.1 (28 Nov 2008) - draft

Projects using OpenLyrics

This is an (incomplete) alphabetic list of projects that use OpenLyrics or are experimenting with using it as their import/export format.

- ChangingSong: <http://changingsong.sf.net/>
- OpenLP: <http://openlp.org/>

Index

C

categories, [7](#)
CCLI support, [7](#)
chords, [7](#)
comments in lyrics, [7](#)

D

date of song release, [7](#)

F

format version, [7](#)

K

keywords for searching, [7](#)

L

last modification time, [7](#)
lines of text, [7](#)

M

metadata, [31](#)
multiple authors, [7](#)
multiple categories, [7](#)
multiple song titles, [7](#)
multiple user-defined items, [7](#)
music properties, [7](#)

N

namespace, [7](#)

P

parts, [7](#)

S

slides, [7](#)
song book, [7](#)
song metadata, [7](#)
song properties, [31](#)
song translator, [7](#)
song variant, [7](#)
song version, [7](#)

T

tagging verse type, [7](#)
theme, [31](#)
translated lyrics, [7](#)
translated song title, [7](#)
translated theme, [7](#)
transliterated lyrics, [7](#)
transliterated song title, [7](#)
transliterated theme, [7](#)
transposition, [7](#)

U

user-defined item, [7](#)

V

verse order, [7](#)