

# Audovia Documentation

## 1 Introduction

**Audovia** is a database application for making music on your Microsoft Windows or Linux Ubuntu laptop or PC. Songs can have up to 15 instrumental voices and a percussion track. Instruments can be chosen from either the default soundbank of 128 instruments or other soundbanks of your choice.

Songs can be developed, tested and edited very quickly and easily by virtue of the database structure and the **JFugue** MusicString notation. Notes within a MusicString are specified by their name and octave or by their MIDI value and their durations are specified either by character code, or numerically. You can use notes from C0 to G10, corresponding to MIDI values 0 to 127. Middle C is C5. Notes can be entered manually or by picking from graphic Treble, Alto, Tenor and Bass staves within the MusicString editor.

The opening phrase of Joy to the World can be written as:

```
C6h B5q. A5i G5h. F5q E5h D5h C5h.
```

Sequences of MusicStrings can be assembled into Patterns, which can be nested to any level. Songs can be constructed from Patterns either timewise or by voice.

The *File/Template* menu item creates song templates by voice where each voice is a Pattern containing other Patterns and MusicStrings. Bars can be MusicStrings or Patterns. Pattern bars can be used to contain separate MusicStrings and/or Patterns.

All MusicStrings and Patterns within a song are given unique names which makes it easy to keep track of them within a composition. A MusicString can be shared between any of the Patterns in a song and it only needs to be edited once for the changes to be effective wherever that MusicString occurs within the song.

You can collaborate with your colleagues on a song by using a MySQL shared database. You can also share your songs by exporting and importing to and from XML files.

**Audovia** will play back your music and also export to MIDI, MusicXML and WAV files for music processing and music publishing. The WAV files can be opened in **Audacity**, then exported to MP3.

### 1.1 Background

**Audovia** is written in Java with a Swing forms user interface. It uses Apache Derby as the default database with an embedded JDBC driver. The music is generated by **JFugue**, a Java API for music programming.

# Audovia Documentation

---

## 2 Installation

Before running **Audovia** on Microsoft Windows you will first need to install the latest version of Java from **Java.com** or, if you are using Linux Ubuntu, you will need to execute the following commands to install Oracle's Java.

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
```

For best results on Linux Ubuntu, we recommend using the Gnome Desktop Environment.

Then, from **Audovia.com**, follow the appropriate link to download and install the desktop shortcut for running **Audovia**.

### 2.1 Soundbanks

**Audovia** uses the Gervill synthesizer to generate sound. Gervill can use the default soundbank or any soundbank with a *.sf2* or *.dls* extension.

Soundbanks can be kept on file or you can use *Soundbanks/Manage Soundbanks* to upload these to the database.

If you play a Pattern and specify the default soundbank, Gervill will search the locations:

- *<Java Runtime location>\lib\audio\\*.dls and \*.sf2*
- *<Windows location>\System32\drivers\gm.dls*

If a soundbank is not found, Gervill will generate one using software.

*You can create your own soundbanks by recording instrument sounds in **Audacity** and importing them into the **Viena** SoundFont editor which is available from the **SynthFont** website.*

### 2.2 Backing up your Songs

Use *File/XML Export* to export a song to an XML file or *File/XML Import* to import a song from an XML file. The entire song structure is exported or imported with the exception of any soundbank reference.

### 2.3 Shared Database (optional)

If you are using a Database as a Service provider, then your hostname, port, database, username and password may be set up for you. In this case, use *Database/Database Connections* to create a new connection and enter the Connection Details. Please note that when entering the Connection Details, there should be no leading or trailing spaces in any of the fields as this will cause a communications failure. Then use *File/Create Tables* to create the database tables on the remote host.

## Audovia Documentation

---

If you are using MySQL as your shared database, we recommend that you create a user in your database called 'guest' with *EXECUTE* as the only privilege. Connecting as 'guest' enables read access to all songs and also the ability to create and log in to individual user accounts for the creation of songs. Full access to these songs can optionally be shared with selected other user accounts.

### 3 Tutorial

**Audovia** opens with the **Songs** form for managing your songs. If you are using numeric note durations within a song, use the **/durations** column to specify whether these are expressed as decimal values of a whole note or as pulses, where 32 pulses represents a quarter note.

To ensure that the different voices in a song are synchronized, all note durations should equate to a whole number of pulses.

Numeric durations should be used for triplets. A quarter note divided into three triplets should be given durations of 11, 11 and 10 pulses, or 12, 10 and 10 pulses if you want to give more emphasis to the first triplet.

To edit a cell, either select the cell and type something into it or double-click on the cell. To copy data in a cell, highlight the text to be copied and press Ctrl-C. To paste, press Ctrl-V. This only works if the cell is in edit mode (yellow).

The **Strings** button opens a form for defining the **JFugue** MusicStrings in a selected song and the **Patterns** button opens a form for declaring the Patterns in the song.

The naming of MusicStrings is independant from the naming of Patterns. You may find it helpful to name MusicStrings in lower case and Patterns with an initial capital.

Use *File/Template* to give you a start when creating new songs.

Use *File/Tree View* to display the structure of Patterns and MusicStrings in a selected song (see below).

#### 3.1 Strings

The **Strings** form is used to define the MusicStrings in the selected song. MusicString notation is described in the next section.

MusicStrings can be edited within this form or, if they are more than two lines long, the **Popup Editor** can be used to provide an editing window.

MusicStrings can be imported from other songs by using *File/Import Strings*. Libraries of MusicStrings can be built and used in this way.

## Audovia Documentation

---

### 3.2 Patterns

The **Patterns** form is used to declare the Patterns in the selected song. A Pattern is a container for other Patterns and MusicStrings.

The **Components** button opens a form for picking the Patterns and MusicStrings that are to be contained within a selected Pattern.

The **Play** button will play the selected Pattern.

The *File* options are *Export to MIDI*, *Export to MusicXML*, *Export to WAV* and *Clone*. If you want to create a Pattern that is very similar to an existing Pattern, use *Clone* and edit the clone.

#### 3.2.1 Example of Cloning

Suppose you have a song which is 128 bars long and the first 12 bars are repeated with a different 12th bar on the repeat.

Use the template to create a song where the voices have 2 parts with 128 bars per part. Clone each Part 1 and call this the Repeat. Insert the Repeat after Part 1 in the voice patterns. Delete bar 13 from Part 1 and delete bar 12 from the Repeat. Bar 13 in the Repeat becomes the different 12th bar.

You can keep the existing bar numbers and since bar numbers are independant for each part you can start Part 2 at bar 13.

### 3.3 Components

The **Components** form is used to pick the Patterns and MusicStrings that are to be contained within the selected Pattern and to specify their position in the sequence in which they are to be played. Picking is done via a drop down list which is displayed when you click on a **Component** cell.

The **Insert** and **Renumber** buttons can be used to insert components into an existing sequence.

The **Drill Down** button allows you to drill down to the child components of a selected Pattern. Drilling down on a MusicString will provide a window for editing that MusicString. When selecting a component for **Drill Down** it is advisable to click on the **Position** cell otherwise the component pick list will be displayed.

If you would like to enter an anonymous MusicString into your Pattern, leave the **Component** cell blank and type the MusicString into the **Anonymous String** cell. This column can be made wider by dragging the boundary between the **Component** and **Anonymous String** headers.

### 3.4 Tree View

The tree view may be used to display the structure of Patterns and MusicStrings in a selected song.

Clicking on a MusicString will open a popup window for editing that MusicString. Use the **Refresh** button to propagate any changes made here throughout the tree.

## Audovia Documentation

---

Clicking on a Pattern will enable the **Play** button and also the *File* options: *Export to MIDI*, *Export to MusicXML*, *Export to WAV* and *Clone*.

### 3.5 MusicXML

If you export a Pattern to a MusicXML file, suitable for input to a music publishing system such as **MuseScore**, then for best results:

- Tempo should be the first token in the Pattern and each different Voice token should appear only once in the Pattern and be followed by one each of Instrument and Key tokens.
- Combined harmony and melody should be represented by separate voices. Notes connected to chords using the \_ character will not be exported to the MusicXML file.
- Bar lines should be used.
- If numeric note durations are used they should match exactly to durations within the range whole note going down by halves to 128th note (dotted or undotted, but not a dotted 128th note).

## 4 MusicStrings

### 4.1 Notes

A C Major scale of quarter notes, starting at middle C, can be written as:

```
C5q D5q E5q F5q G5q A5q B5q C6q
```

or as:

```
C5/0.25 D5/0.25 E5/0.25 F5/0.25 G5/0.25 A5/0.25 B5/0.25 C6/0.25
```

or, if you have **/durations** set to pulses, as:

```
C5/32 D5/32 E5/32 F5/32 G5/32 A5/32 B5/32 C6/32
```

In addition to the note letters, A to G, you can use R for a rest. Sharps, flats and naturals can be added by placing the character #, b or n immediately after the note letter so B-flat above middle C is written as Bb5.

Please note that, if you are transcribing music, accidentals in **Audovia** apply only to the immediately following note and not to the end of the bar as in conventional music notation.

MusicStrings can optionally be split into bars (or measures) by using the vertical bar character (|):

```
C5q D5q E5q F5q | G5q A5q B5q C6q |
```

As an alternative to note letters, MIDI values, enclosed in square brackets, may be used:

```
[60]q [62]q [64]q [65]q | [67]q [69]q [71]q [72]q |
```

## Audovia Documentation

---

### 4.1.1 Durations

The duration characters are:

w	whole note
h	half note
q	quarter note
i	eighth note
s	sixteenth note
t	thirty-second note
x	sixty-fourth note
o	one-twenty-eighth note

Dotted duration can be achieved by putting the period character (.) immediately after the duration character.

### 4.1.2 Chords

Chords are formed by adding the constituent notes together. A C Major chord can be written as:

C5q+E5q+G5q

### 4.1.3 Ties

Two or more notes of the same pitch can be tied together by using the hyphen character (-). Place the hyphen immediately after the duration of the note at the start of the tie and immediately before the duration of the note at the end of the tie. Notes in the middle of the tie have hyphens placed immediately before and after the note duration:

C5q D5q E5q F5q- | F5-w- | F5-q G5q A5q B5q |

## 4.2 Tempo

A tempo of 120 beats per minute can be expressed as:

T120

or as:

T[allegro]

Note the use of a pre-defined numeric constant within the square brackets. More tempo constants are available from *Insert/Tempo*.

## 4.3 Constants

Constants are defined using the \$ character followed by the constant name.

## Audovia Documentation

---

### 4.3.1 Numeric Constants

Numeric constants can be used anywhere that a number would appear in a MusicString. In addition to the pre-defined constants available from the *Insert* menu, you can define your own constants. For example, a scale with non-standard MIDI values could be defined as:

```
$A1=70
$G=68
$F=67
$E=65
$D=63
$C=62
$B=60
$A=58
```

and played as:

```
[A]q [B]q [C]q [D]q [E]q [F]q [G]q [A1]q
```

### 4.3.2 String Constants

Suppose you wanted to use the following arpeggio several times in your music:

```
F3i A3i C4i F4i C4i A3i
```

You could define a string constant as:

```
$arpeggioFoctave3=F3i~A3i~C4i~F4i~C4i~A3i
```

Then, in your music, you could refer to it as:

```
{arpeggioFoctave3}
```

Note the use of curly brackets for string constants.

## 4.4 Voices

Voices are specified by the V character followed by a number from 0 to 15. Note that V9 is the percussion voice and has its own set of instruments.

## 4.5 Key Signatures

Key signatures are specified by the K character followed by a note letter (or a note letter followed by # or b) followed by maj or min to indicate a major or minor scale so the key of G Major is written as KGmaj.

## 4.6 Instruments

Instruments are specified by the I character followed by a number from 0 to 127. You can use *Insert/Instrument* to pick one of the pre-defined values.

## Audovia Documentation

---

### 4.7 MIDI Controller

MIDI controller events can be specified by the X character followed by the controller number followed by the equals sign (=) followed by a value. You can use *Insert/Controller* to pick one of the pre-defined controllers.

### 4.8 Pitch Wheel

A change of pitch can be specified by the & character followed by a number from 0 to 16383. This affects all following notes:

&0	lowers the pitch by a full tone;
&8192	returns the pitch to no change;
&16383	raises the pitch by a full tone.

## 5 Linux Ubuntu

On Linux Ubuntu, the desktop shortcut will be installed in */usr/share/applications*.

**Audovia** uses Liberation Fonts. If these are not already installed, search for fonts-liberation in the Ubuntu Software Center.

Since there is no default soundbank in Ubuntu, Gervill will generate one automatically. You may wish to search the Internet for SoundFont files to get better quality. You can also follow the SoundFont links on the **SynthFont** website.

A quick way to get a SoundFont is to first run *sudo apt-get install timidity lame* from the command prompt, then from the Ubuntu Software Center, find timidity and select the *Fluid (R3) General MIDI SoundFont (GM)* Add-on. This will put a SoundFont in */usr/share/sounds/sf2*. To use this, select Soundbank from Filewhen playing a Pattern.

To convert your songs to MP3, you can either *Export to WAV* then open the WAV files in **Audacity**, then export to MP3, or you can create a *<config.cfg>* file with the following entry:

```
soundfont <soundfontfile.sf2>
```

Then *Export to MIDI* and, from the command prompt, run:

```
timidity <song.mid> -c <config.cfg> -Ow -o - | lame - <song.mp3>
```

Please note that timidity will remove any silence from the beginning of the song. If you want to restore some silence, use the following commands:

```
timidity <song.mid> -c <config.cfg> -Ow -o <song.wav>
sox <song.wav> <song_padded.wav> pad <seconds>
lame <song_padded.wav> <song.mp3>
```

To apply reverb, open the .wav file in **Audacity** and use the *Reverb* effect, then *File/Export* to MP3.



## Audovia Documentation

---

### 6 Predefined Constants

#### 6.1 Instrument Names

PIANO	0
ACOUSTIC_GRAND	0
BRIGHT_ACOUSTIC	1
ELECTRIC_GRAND	2
HONKEY_TONK	3
ELECTRIC_PIANO	4
ELECTRIC_PIANO_1	4
ELECTRIC_PIANO_2	5
HARPISCHORD	6
HARPSICHORD	6
CLAVINET	7
CELESTA	8
GLOCKENSPIEL	9
MUSIC_BOX	10
VIBRAPHONE	11
MARIMBA	12
XYLOPHONE	13
TUBULAR_BELLS	14
DULCIMER	15
DRAWBAR_ORGAN	16
PERCUSSIVE_ORGAN	17
ROCK_ORGAN	18
CHURCH_ORGAN	19
REED_ORGAN	20
ACCORDIAN	21
HARMONICA	22
TANGO_ACCORDIAN	23
GUITAR	24
NYLON_STRING_GUITAR	24
STEEL_STRING_GUITAR	25
ELECTRIC_JAZZ_GUITAR	26
ELECTRIC_CLEAN_GUITAR	27
ELECTRIC_MUTED_GUITAR	28
OVERDRIVEN_GUITAR	29
DISTORTION_GUITAR	30
GUITAR_HARMONICS	31
ACOUSTIC_BASS	32
ELECTRIC_BASS_FINGER	33
ELECTRIC_BASS_PICK	34
FRETLESS_BASS	35
SLAP_BASS_1	36
SLAP_BASS_2	37
SYNTH_BASS_1	38
SYNTH_BASS_2	39
VIOLIN	40
VIOLA	41

## Audovia Documentation

---

CELLO	42
CONTRABASS	43
TREMOLO_STRINGS	44
PIZZICATO_STRINGS	45
ORCHESTRAL_STRINGS	46
TIMPANI	47
STRING_ENSEMBLE_1	48
STRING_ENSEMBLE_2	49
SYNTH_STRINGS_1	50
SYNTH_STRINGS_2	51
CHOIR_AAHS	52
VOICE_OOHS	53
SYNTH_VOICE	54
ORCHESTRA_HIT	55
TRUMPET	56
TROMBONE	57
TUBA	58
MUTED_TRUMPET	59
FRENCH_HORN	60
BRASS_SECTION	61
SYNTHBRASS_1	62
SYNTH_BRASS_1	62
SYNTHBRASS_2	63
SYNTH_BRASS_2	63
SOPRANO_SAX	64
ALTO_SAX	65
TENOR_SAX	66
BARITONE_SAX	67
OBOE	68
ENGLISH_HORN	69
BASSOON	70
CLARINET	71
PICCOLO	72
FLUTE	73
RECORDER	74
PAN_FLUTE	75
BLOWN_BOTTLE	76
SKAKUHACHI	77
WHISTLE	78
OCARINA	79
LEAD_SQUARE	80
SQUARE	80
LEAD_SAWTOOTH	81
SAWTOOTH	81
LEAD_CALLIOPE	82
CALLIOPE	82
LEAD_CHIFF	83
CHIFF	83
LEAD_CHARANG	84
CHARANG	84
LEAD_VOICE	85

## Audovia Documentation

---

VOICE	85
LEAD_FIFTHS	86
FIFTHS	86
LEAD_BASSLEAD	87
BASSLEAD	87
PAD_NEW_AGE	88
NEW_AGE	88
PAD_WARM	89
WARM	89
PAD_POLYSYNTH	90
POLYSYNTH	90
PAD_CHOIR	91
CHOIR	91
PAD_BOWED	92
BOWED	92
PAD_METALLIC	93
METALLIC	93
PAD_HALO	94
HALO	94
PAD_SWEEP	95
SWEEP	95
FX_RAIN	96
RAIN	96
FX_SOUNDTRACK	97
SOUNDTRACK	97
FX_CRYSTAL	98
CRYSTAL	98
FX_ATMOSPHERE	99
ATMOSPHERE	99
FX_BRIGHTNESS	100
BRIGHTNESS	100
FX_GOBLINS	101
GOBLINS	101
FX_ECHOES	102
ECHOES	102
FX_SCI-FI	103
SCI-FI	103
SITAR	104
BANJO	105
SHAMISEN	106
KOTO	107
KALIMBA	108
BAGPIPE	109
FIDDLE	110
SHANAI	111
TINKLE_BELL	112
AGOGO	113
STEEL_DRUMS	114
WOODBLOCK	115
TAIKO_DRUM	116
MELODIC_TOM	117

## Audovia Documentation

---

SYNTH_DRUM	118
REVERSE_CYMBAL	119
GUITAR_FRET_NOISE	120
BREATH_NOISE	121
SEASHORE	122
BIRD_TWEET	123
TELEPHONE_RING	124
HELICOPTER	125
APPLAUSE	126
GUNSHOT	127

### 6.2 Percussion Names

ACOUSTIC_BASS_DRUM	35
BASS_DRUM	36
SIDE_STICK	37
ACOUSTIC_SNARE	38
HAND_CLAP	39
ELECTRIC_SNARE	40
LOW_FLOOR_TOM	41
CLOSED_HI_HAT	42
HIGH_FLOOR_TOM	43
PEDAL_HI_HAT	44
LOW_TOM	45
OPEN_HI_HAT	46
LOW_MID_TOM	47
HI_MID_TOM	48
CRASH_CYMBAL_1	49
HIGH_TOM	50
RIDE_CYMBAL_1	51
CHINESE_CYMBAL	52
RIDE_BELL	53
TAMBOURINE	54
SPLASH_CYMBAL	55
COWBELL	56
CRASH_CYMBAL_2	57
VIBRASLAP	58
RIDE_CYMBAL_2	59
HI_BONGO	60
LOW_BONGO	61
MUTE_HI_CONGA	62
OPEN_HI_CONGA	63
LOW_CONGA	64
HIGH_TIMBALE	65
LOW_TIMBALE	66
HIGH_AGOGO	67
LOW_AGOGO	68
CABASA	69
MARACAS	70

## Audovia Documentation

---

SHORT_WHISTLE	71
LONG_WHISTLE	72
SHORT_GUIRO	73
LONG_GUIRO	74
CLAVES	75
HI_WOOD_BLOCK	76
LOW_WOOD_BLOCK	77
MUTE_CUICA	78
OPEN_CUICA	79
MUTE_TRIANGLE	80
OPEN_TRIANGLE	81

### 6.3 Controller Names

BANK_SELECT_COARSE	0
MOD_WHEEL_COARSE	1
BREATH_COARSE	2
FOOT_PEDAL_COARSE	4
PORTAMENTO_TIME_COARSE	5
DATA_ENTRY_COARSE	6
VOLUME_COARSE	7
BALANCE_COARSE	8
PAN_POSITION_COARSE	10
EXPRESSION_COARSE	11
EFFECT_CONTROL_1_COARSE	12
EFFECT_CONTROL_2_COARSE	13
SLIDER_1	16
SLIDER_2	17
SLIDER_3	18
SLIDER_4	19
BANK_SELECT_FINE	32
MOD_WHEEL_FINE	33
BREATH_FINE	34
FOOT_PEDAL_FINE	36
PORTAMENTO_TIME_FINE	37
DATA_ENTRY_FINE	38
VOLUME_FINE	39
BALANCE_FINE	40
PAN_POSITION_FINE	42
EXPRESSION_FINE	43
EFFECT_CONTROL_1_FINE	44
EFFECT_CONTROL_2_FINE	45
HOLD_PEDAL	64
HOLD	64
PORTAMENTO	65
SUSTENUTO_PEDAL	66
SUSTENUTO	66
SOFT_PEDAL	67
SOFT	67

## Audovia Documentation

---

LEGATO_PEDAL	68
LEGATO	68
HOLD_2_PEDAL	69
HOLD_2	69
SOUND_VARIATION	70
VARIATION	70
SOUND_TIMBRE	71
TIMBRE	71
SOUND_RELEASE_TIME	72
RELEASE_TIME	72
SOUND_ATTACK_TIME	73
ATTACK_TIME	73
SOUND_BRIGHTNESS	74
BRIGHTNESS	74
SOUND_CONTROL_6	75
CONTROL_6	75
SOUND_CONTROL_7	76
CONTROL_7	76
SOUND_CONTROL_8	77
CONTROL_8	77
SOUND_CONTROL_9	78
CONTROL_9	78
SOUND_CONTROL_10	79
CONTROL_10	79
GENERAL_PURPOSE_BUTTON_1	80
GENERAL_BUTTON_1	80
BUTTON_1	80
GENERAL_PURPOSE_BUTTON_2	81
GENERAL_BUTTON_2	81
BUTTON_2	81
GENERAL_PURPOSE_BUTTON_3	82
GENERAL_BUTTON_3	82
BUTTON_3	82
GENERAL_PURPOSE_BUTTON_4	83
GENERAL_BUTTON_4	83
BUTTON_4	83
EFFECTS_LEVEL	91
EFFECTS	91
TREMULO_LEVEL	92
TREMULO	92
CHORUS_LEVEL	93
CHORUS	93
CELESTE_LEVEL	94
CELESTE	94
PHASER_LEVEL	95
PHASER	95
DATA_BUTTON_INCREMENT	96
DATA_BUTTON_INC	96
BUTTON_INC	96
DATA_BUTTON_DECREMENT	97
DATA_BUTTON_DEC	97

## Audovia Documentation

---

BUTTON_DEC	97
NON_REGISTERED_COARSE	98
NON_REGISTERED_FINE	99
REGISTERED_COARSE	100
REGISTERED_FINE	101
ALL_SOUND_OFF	120
ALL_CONTROLLERS_OFF	121
LOCAL_KEYBOARD	122
ALL_NOTES_OFF	123
OMNI_MODE_OFF	124
OMNI_OFF	124
OMNI_MODE_ON	125
OMNI_ON	125
MONO_OPERATION	126
MONO	126
POLY_OPERATION	127
POLY	127

### 6.4 Combined Controller Names

```
(index = coarse_controller_index * 128 + fine_controller_index)
```

BANK_SELECT	16383
MOD_WHEEL	161
BREATH	290
FOOT_PEDAL	548
PORTAMENTO_TIME	677
DATA_ENTRY	806
VOLUME	935
BALANCE	1064
PAN_POSITION	1322
EXPRESSION	1451
EFFECT_CONTROL_1	1580
EFFECT_CONTROL_2	1709
NON_REGISTERED	12770
REGISTERED	13028

### 6.5 Values for some controllers

ON	127
OFF	0
DEFAULT	64

## Audovia Documentation

---

### 6.6 Tempo Values

GRAVE	40
LARGO	45
LARGHETTO	50
LENTO	55
ADAGIO	60
ADAGIETTO	65
ANDANTE	70
ANDANTINO	80
MODERATO	95
ALLEGRETTO	110
ALLEGRO	120
VIVACE	145
PRESTO	180
PRETISSIMO	220



## Audovia Documentation

---

### 7 XML Specification

Here is the XML specification of the *.sbxml* files used in *File/XML Export* and *File/XML Import*.

```
<?xml version="1.0"?>

<songs>

  <song>

    <song_name>Song Name</song_name>
    <numeric_duration_type>decimal or pulses</numeric_duration_type>

    <components>

      <component>

        <component_type>pattern or string</component_type>
        <component_name>Pattern or String Name</component_name>
        <string_value>a JFugue MusicString</string_value>

      </component>

    </components>

    <pattern_components>

      <pattern_component>

        <pattern_name>Pattern Name</pattern_name>
        <component_position>an integer value</component_position>
        <component_type>pattern or string</component_type>
        <component_name>Pattern or String Name</component_name>
        <anonymous_string>a JFugue MusicString</anonymous_string>

      </pattern_component>

    </pattern_components>

  </song>

</songs>
```