**MML To MIDI Language Specification**

**Introduction**

The music macro language (MML) is a music description language that has been in use since 1978, although this was an early version. There has never been an official specification, so each implementation varies slightly, and over the years the language has evolved. The MML to MIDI converter uses a version of the MML derived largly from “Classical MML” with some “Modern MML” features present. Some new specific commands are included also, and some commands are changed where necessary. This document outlines this version of the MML used by the program, and contrasts it to existing syntax where it significantly differs.

**Comments**

Comments are started with two hash characters, “##”. This makes the remainder of the line a comment; any more hashes found on the line have no effect. Two hashes are used because single hash starts a meta command.

**Playing Notes**

Playing notes is done by starting a line with the word “music”, then entering the notes after a space. Any spaces aside from this initial one are ignored, and the command is ended with a new line. An example of the music command is:

music c5e5g5

**Note Syntax**

Notes are written as the note name followed by the length of the note as a digit – each value for this digit represents a musical note length, which can be seen in the table below. A rest is represented by the note name “r”. To play an accidental note a “+” or “-”, respectively, is added after the note name and before the note length. Accidentals applied to rests are ignored.

|  |  |  |
| --- | --- | --- |
| MML Note Value Number | Musical Note | |
| Relative Value | Name |
| 0 | 1/32 | Demisemiquaver |
| 1 | 1/16 | Semiquaver |
| 2 | 1/16 + 1/32 | Dotted semiquaver |
| 3 | 1/8 | Quaver |
| 4 | 1/8 + 1/16 | Dotted quaver |
| 5 | 1/4 | Crochet |
| 6 | 1/4 + 1/8 | Dotted crochet |
| 7 | 1/2 | Minim |
| 8 | 1/2 + 1/4 | Dotted minium |
| 9 | 1 | Semibreve |

To alter how each note is played, there are some of commands entered with the notes. These are listed below:

* o[digit] – set the octave each following note is played in. The number represents the scientific pitch notation (SPN) number of the desired octave. All notes entered before this command is entered are played in the 4th SPN octave (“A” will be 440 Hz.)
* v[digit] – the v command sets the volume of the following notes. A 9 as the digit will set the volume to 100%. A 0 will set the volume to 10%. By default, notes will play at 100% volume.
* l[digit] – set the length of notes where the length is not specified. The digit represents the note length as specified previously.

In modern MML there is also a “t” command that sets the tempo. This is not included because MIDI only supports one setting of the tempo for an entire song.

**Meta Commands**

These commands are entered on their own lines only once and are all preceded by a single hash. These tell the converter how the entire file should be played and add information to the MIDI file. They are listed below:

* #tempo [BPM] – set the tempo in BPM of the track (where a beat is a crochet.) BPM is used as po This should be set the same in each MML track file when combining them into one MIDI file. If multiple tempo commands are found in the file, only the latest one will be used. The default tempo is 120 BPM.
* #instrument [general MIDI patch number] – set the instrument the file should be played with. If multiple instances of this command are found, only the latest will be used. The default instrument is a piano (GM patch number 0.)
* #name [name] – set the name of the track. This is put verbatim into the MIDI file in a track name meta event.

**Macros**

A macro in this version of MML is a statement that tells the converter to replace every instance of a string with another string in the entirety of the program. This is a powerful tool that can make the file more readable and much less verbose. A macro is initialised by writing the string to be replaced as the first word on a line, then an equals character, “=”, and then the replacing string. The string to be replaced cannot contain any numbers or symbols aside from “\_”. An example macro definition is below:

MiddleA = o4a5

Spaces are ignored and the macro is ended with a new line.

**Full Example**

To conclude the document, an example of a valid MML file is shown below.

|  |
| --- |
| #Example comment  name piano0  instrument 0  tempo 120  middle\_c\_scale = o4c5d5e5f5g5a5b5o5c5  play v9middle\_c\_scale |