Time Signature Conversions

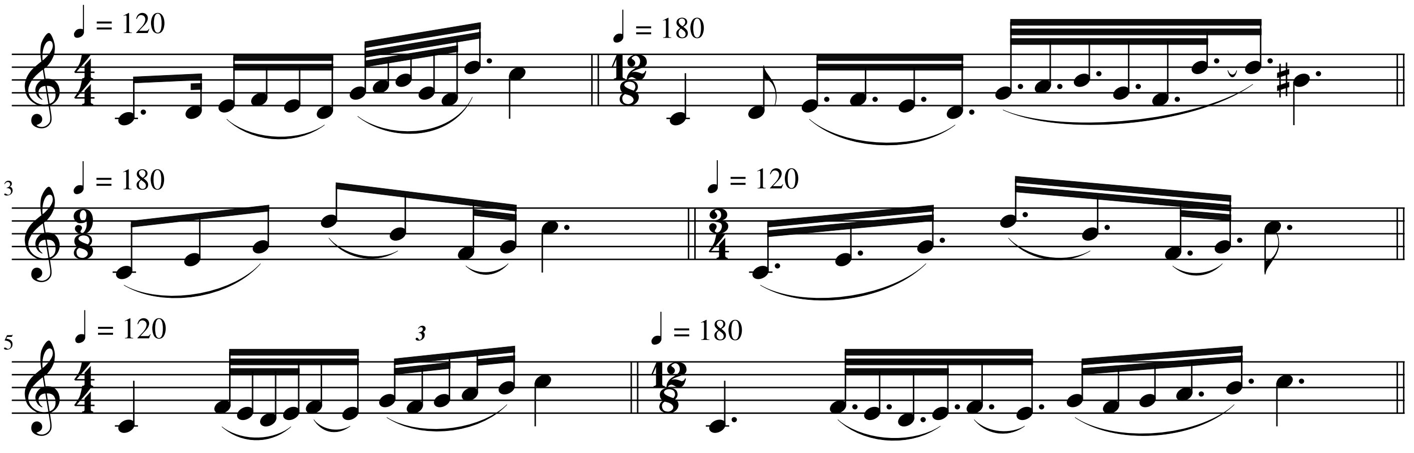
A Tutorial for Converting Between Duple and Triple Meters

Benjamin Robert Tubb

Because **MakeMusic's** *Finale* program uses an internal resolution of 1024 PPQN (parts/pulses per quarter note), its inability to exactly handle subdivisions of the beat by factors other than two becomes problematic in both its internal playback as well as external playback from the duration conversions to/from MIDI files.

The MIDI (Musical Intrument Digital Interface) File Specification by the MMA (MIDI Manufactureres Association) as a minimum requires a 24 PPQN for syncing to MIDI clock bytes when using master/slave control between MIDI devices. Thus all PPQNs used for MIDI files should have 24 as a common facter. The factors of 24 are 2^3 * 3. Unfortunately the 1024 PPQN used by *Finale* is only factorable by 2 (i.e. 1024 = 2^10). Therefore their least common multiple would be 3072 (2^10 * 3). No sequencer suports this, let alone uses 1024 PPQN except *Finale*. The highest resoltions currently supported by some sequencers are 768 (2^8 * 3) and 960 (2^6 * 3 * 5). The *960 family* of PPQNs at least includes the factor of 5 for which its lower resolution values as 120, 240, 480 have been in common use. Likewise the *768 family* of PPQNs includes the lower resolution values 48, 96, 192, and 384. Note that each *family* of PPQNs supports the 24 PPQN resolution as a common factor for their lowest *member's* resolutions, i.e. 48 for the *960 family*, and 120 for the *768 family*. The use of other prime factors as 7, 11 and 13 are the most uncommon of all for practical use. In anycase, I contend that all prime factors above 2 and 3 are *felt* as multiples of 2+3 in various combinations, and thus are not directly needing to be supported (or notated) as *native* factors.

The following are conversions to get *Finale's* files to accurately convert to and from Time Signatures of duple and tuple meters and avoid the PPQN quanitization used by *Finale* to handle triple subdivisions of the beat. To do so, consider the most common case between so called *straight* (duple) and *swing* (tuple) time where 4/4 is *swung* in effect into 12./8 time. To do so accurately for notation, play-back and MIDI file output conversion, if given the music in 4/4, to convert it to 12/8, first triple all durations and then halve them. To then make the original tempo [DT], in number of quarter notes per minute, *convert* to its tuple *counterpart*, take DT and divide it by two, then multiply it by three to get the new [TT] tuple tempo. For example, given 120 BPM, then 120/2 = 60, 60*3 = 180. The reverse process is thus, given TT, divide it by three, then multiply it by two. For example, given 180 BPM, then 180/3 = 60, 60 * 2 = 120.



Duple to Tuple Tempo Conversions

20	->	30	13.33	->	20
30	->	45	26.67	->	40
40	->	60	33.33	->	60
50	->	75	46.67	->	70
60	->	90	53.33	->	80
70	->	105	66.67	->	100
80	->	120	73.33	->	110
90	->	135	86.67	->	130
100	->	150	93.33	->	140
110	->	165	106.67	->	160
120	->	180	113.33	->	170
130	->	195	126.67	->	190
140	->	210	133.33	->	200
150	->	225			
160	->	240			
170	->	255			
180	->	270			
190	->	285			
200	->	300			

Recommended Minimum PPQN

384	Whole
256	Triplet Whole
192	Half
128	Triplet Half
96	**Quarter**
64	Triplet Quarter
48	Eighth
32	Triplet Eighth
24	Sixteenth
16	Triplet Sixteenth
12	Thirtysecond
8	Triplet Thirtysecond
6	Sixtyfourth
4	Triplet Sixtyfourth
3	128th
2	Triplet 128th
1	Triplet 256th

***Finale's* Only Exact Quantized Note Values**

4096	Whole
2048	Half
1024	**Quarter**
512	Eighth
256	Sixteenth
128	Thirysecond
64	Sixtyfourth
32	128th
16	256th
8	512th
4	1024th
2	2048th
1	4096th

Copyright © 2007 (Feb 8) by Benjamin Robert Tubb.
www.pdmusic.org