Appendix A: Programming Language Grammar

Regular Expressions:

<*identifier*>	::= [a-zA-Z][a-zA-Z0-9_]*
<*string*>	::= “.*”
<*integer*>	::= [0-9]+
<*single*>	::= [0-9]*”.”?[0-9]*([Ee][+-]?[0-9]+)?s?
<*double*>	::= [0-9]*”.”?[0-9]*([Ee][+-]?[0-9]+)?d?
<*fixed*>	::= [0-9]*”.”?[0-9]*([Ee][+-]?[0-9]+)?f?

LL(1) Grammar:

<*form*>	::= <*function*> **;** ::= <*prototype*> **;**
<*function*>	::= **func** <*identifier*> **(** <*formalparamstart*> **) :** <*type*> <*expr*>
<*prototype*>	::= **proto** <*identifier*> **(** <*formalparamstart*> **) :** <*type*>
<*formalparamstart*>	::= <*formalparamlist*> ::=
<*type*>	::= **bool** ::= **int** ::= **single** ::= **double** ::= **fixed** ::= **boolarray** ::= **intarray** ::= **singlearray** ::= **doublearray** ::= **fixedarray**
<*expr*>	::= <*expr2*> ::= **if** <*ifrest*> ::= <*whileloop*> ::= **do** <*expr*> <*loopwhileuntil*> ::= <*untilloop*> ::= **set** <*expr*> **:=** <*expr*> ::= **resize** <*expr*> **to** <*expr*> ::= **error** <*string*> **resumable** <*expr*> ::= **getsampleleft** <*string*> ::= **getsampleright** <*string*> ::= **getsample** <*string*> ::= **getwavenumframes** <*string*> ::= **getwavenumtables** <*string*> ::= **getwavedata** <*string*>
<*formalparamlist*>	::= <*formalarg*> <*formalargtail*>
<*whileloop*>	::= **while** <*expr*> **do** <*expr*>
<*untilloop*>	::= **until** <*expr*> **do** <*expr*>
<*vartail*>	::= **=** <*expr*> ::= **(** <*expr*> **)**
<*ifrest*>	::= <*expr*> **then** <*expr*> <*iftail*>
<*loopwhileuntil*>	::= **while** <*expr*> ::= **until** <*expr*>
<*expr2*>	::= <*expr3*> <*expr2prime*>
<*formalarg*>	::= <*identifier*> **:** <*type*>
<*formalargtail*>	::= **,** <*formalparamlist*> ::=
<*iftail*>	::= **else** <*expr*> ::= **elseif** <*ifrest*>

::=

<expr3> ::= *<expr4>* *<expr3prime>*

<expr2prime> ::= *<conj_oper>* *<expr3>* *<expr2prime>*
::=

<expr4> ::= *<expr5>* *<expr4prime>*

<expr3prime> ::= *<rel_oper>* *<expr4>* *<expr3prime>*
::=

<conj_oper> ::= **and**
::= **or**
::= **xor**

<expr5> ::= *<expr6>* *<expr5prime>*

<expr4prime> ::= *<add_oper>* *<expr5>* *<expr4prime>*
::=

<rel_oper> ::= **<**
::= **<=**
::= **>**
::= **>=**
::= **=**
::= **<>**

<expr6> ::= *<unary_oper>* *<expr6>*
::= *<expr7>*

<unary_oper> ::= **not**
::= **sin**
::= **cos**
::= **tan**
::= **asin**
::= **acos**
::= **atan**
::= **ln**
::= **exp**
::= **bool**
::= **int**
::= **single**
::= **double**
::= **fixed**
::= **sqr**
::= **sqrt**
::= **abs**
::= **-**
::= **sign**
::= **length**

<expr5prime> ::= *<mult_oper>* *<expr6>* *<expr5prime>*
::=

<add_oper> ::= **+**
::= **-**

<expr7> ::= *<expr8>* *<expr7prime>*

<mult_oper> ::= ***
::= */*
::= **div**
::= **mod**
::= **<<**
::= **>>**

<expr8> ::= *<identifier>*
::= *<integer>*
::= *<single>*
::= *<double>*
::= *<fixed>*
::= *<string>*
::= **true**
::= **false**
::= **(** *<exprlist>* **)**

<*expr7prime*> ::= <*arraysubscript*>
::= <*funccall*>
::= <*exponentiation*>
::=

<*funccall*> ::= **(** <*actualstart*> **)**

<*arraysubscript*> ::= **[** <*exprlist*> **]**

<*exprlist*> ::= <*exprlistelem*> <*exprlisttail*>

<*exponentiation*> ::= **^** <*expr7*>

<*actualstart*> ::= <*actuallist*>
::=

<*actuallist*> ::= <*expr*> <*actualtail*>

<*actualtail*> ::= **,** <*actuallist*>
::=

<*exprlisttail*> ::= **;** <*exprlist*>
::=

<*exprlistelem*> ::= <*expr*>
::= **var** <*identifier*> **:** <*type*> <*vartail*>
::= <*prototype*>

Appendix B: Instrument Definition Grammar

Terminals:

<*identifier*> ::= ([a-zA-Z][a-zA-Z0-9_]*) | (“.*”)
<*integer*> ::= [0-9]+
<*float*> ::= [0-9]*”.”?[0-9]*([Ee][+-]?[0-9]+)?
<*boolean*> ::= true | false

LL(1) Grammar:

<*instr_definition*> ::= **instrument (** <*instr_list*> **)**

<*instr_list*> ::= <*instr_elem*> **;** <*instr_list*>
::=

<*instr_elem*> ::= **loudness** <*number*>
::= **frequencylfo (** <*lfo_definition*> **)**
::= **oscillator** <*identifier*> **(** <*oscillator_definition*> **)**

<*number*> ::= <*integer*>
::= <*float*>

<*lfo_definition*> ::= <*lfo_elem*> **;** <*lfo_definition*>
::=

<*oscillator_definition*> ::= <*oscillator_elem*> **;** <*oscillator_definition*>
::=

<*lfo_elem*> ::= **freqenvelope (** <*envelope_definition*> **)**
::= **ampenvelope (** <*envelope_definition*> **)**
::= **oscillator** <*oscillator_type*>
::= **modulation** <*modulation_type*>
::= **exponential**
::= **linear**

<*oscillator_type*> ::= **constant**
::= **signsine**
::= **plussine**
::= **signtriangle**
::= **plustriangle**
::= **signsquare**
::= **plussquare**
::= **signramp**

::= **plusramp**
::= **signlinfuzz**
::= **pluslinfuzz**
::= **wavetable (** <*samplelist_definition*> **) (** <*envelope_definition*> **)**

<*modulation_type*> ::= **additive**
::= **multiplicative**
::= **inversemult**

<*oscillator_elem*> ::= **type** <*oscillator_type*>
::= **samplelist (** <*samplelist_definition*> **)**
::= **modulators (** <*modulator_list*> **)**
::= **loudness** <*number*>
::= **freqmultiplier** <*number*>
::= **freqdivisor** <*integer*>
::= **freqadder** <*number*>
::= **makeoutput** <*boolean*>
::= **loudnessenvelope (** <*envelope_definition*> **)**
::= **loudnesslfo (** <*lfo_definition*> **)**
::= **indexenvelope (** <*envelope_definition*> **)**
::= **indexlfo (** <*lfo_definition*> **)**
::= **stereobias** <*number*>
::= **displacement** <*number*>

<*oscillator_type*> ::= **sampled**
::= **wavetable**

<*envelope_definition*> ::= <*envelope_elem*> **;** <*envelope_definition*>
::=

<*samplelist_definition*> ::= <*samplelist_elem*> **;** <*samplelist_definition*>
::=

<*modulator_list*> ::= <*modulator_elem*> **;** <*modulator_list*>
::=

<*envelope_elem*> ::= **totalscaling** <*number*>
::= **points (** <*env_point_list*> **)**

<*samplelist_elem*> ::= <*identifier*> <*number*>

<*modulator_elem*> ::= **source** <*identifier*> **scale** <*modulation_dynamic*> **originadjust**
<*modulation_dynamic*> **type** <*modulator_types*> **target** <*target_type*>

<*modulation_dynamic*> ::= **envelope (** <*envelope_definitions*> **)** <*mod_dyn_list*>

<*mod_dyn_list*> ::= **lfo (** <*lfo_definitions*> **)** <*mod_dyn_list*>
::=

<*modulator_types*> ::= **additive**
::= **multiplicative**

<*target_type*> ::= **phasegen**
::= **output**

<*env_point_list*> ::= <*env_point_elem*> **;** <*env_point_list*>
::=

<*env_point_elem*> ::= **delay** <*number*> **level** <*number*> <*env_attributes*>
::= **delay** <*number*> **scale** <*number*> <*env_attributes*>
::= **origin**

<*env_attributes*> ::= <*env_one_attribute*> <*env_attributes*>
::=

<*env_one_attribute*> ::= **sustainpoint** <*integer*>
::= **releasepoint** <*integer*>
::= **sustainpointnoskip** <*integer*>
::= **releasepointnoskip** <*integer*>
::= **ampaccent1** <*number*>
::= **ampaccent2** <*number*>
::= **ampaccent3** <*number*>
::= **ampaccent4** <*number*>
::= **ampfreq** <*number*> <*number*>
::= **rateaccent1** <*number*>
::= **rateaccent2** <*number*>

::= **rateaccent3** <*number*>
::= **rateaccent4** <*number*>
::= **ratefreq** <*number*> <*number*>
::= **exponential**
::= **linear**

Appendix C: Grammar for File Format

This section describes version 1 of the file format used by *Out Of Phase*. The file format is hierarchical, and points where a sub-block is inserted are denoted in italics.

General Information Subblock Structure:

4-byte file format version code
"Syn1" - first file format
1-byte unsigned tab size code
should be in the range of 1..255
4-byte little endian comment text length (positive 2s complement, in bytes)
n-byte character data for comment text (line feed = 0x0a)
1-byte stereo playback flag
0 = mono
1 = stereo
1-byte surround encoding flag
0 = no surround encoding
4-byte little endian output sampling rate
should be in the range of 100..65535
4-byte little endian envelope update rate
should be in the range of 1..65535
4-byte little endian large integer coded decimal beats per minute
large integer coded decimal is decimal * 1000000 with a
range of -1999.999999 to 1999.999999
4-byte little endian large integer coded decimal total volume scaling factor
1-byte number of bits to output
should be 8, 16, 24, or 32
1-byte flag for interpolation over time
0 = don't interpolate over time
1 = do interpolate over time (when resampling waveforms)
1-byte flag for interpolation across waves
0 = don't interpolate across waves
1 = do interpolate across waves (when wave table synthesis index is
not an integer)
4-byte little endian large integer coded decimal scanning gap
4-byte little endian large integer coded decimal buffer duration (in seconds)
1-byte flag for clipping warning
0 = don't warn about clipped samples
1 = do warn about clipped samples
1-byte flag for song post processing enabling
0 = don't do song postprocessing
1 = do song postprocessing
4-byte little endian length of song post processing function
n-bytes of post processing function text (line fed = 0x0a)

4-bytes little endian number of sample objects (positive 2s complement)
n-bytes of data for the sample objects (see below)

4-byte little endian function object count (positive 2s complement)
n-bytes of data for the function objects (see below)

4-byte little endian number of algorithmic sample objects (positive 2s complement)
n-byte data for the algorithmic sample objects (see below)

4-byte little endian number of wave table objects (positive 2's complement)
n-bytes data for the wave table objects (see below)

4-byte little endian number of algorithmic wave table objects (positive 2s complement)
n-byte data for all the algorithmic wave table objects (see below)

4-byte little endian number of instrument objects (positive 2s complement)
n-byte data for the instrument objects (see below)

4-byte little endian number of tracks (positive 2s complement)
n-bytes data for the track objects (see below)
n-bytes data for background display information (a chunk for each track, see below)

Sample Object Subblock Structure:

1-byte sample version number
  should be 1
2-byte little endian window X position (signed, origin at top-left corner)
2-byte little endian window Y position
2-byte little endian window width
2-byte little endian window height
4-byte little endian sample name length descriptor (positive 2's complement)
n-byte sample name text (line feed = 0x0a)
4-byte little endian sample formula length descriptor (positive 2's complement)
n-byte sample formula text (line feed = 0x0a)
4-byte little endian sample frame index of sample's origin
4-byte little endian sample frame index of loop 1 start
  must be a valid index, i.e. >= 0 and < num sample frames
4-byte little endian sample frame index of loop 1 end
  must be a valid index, i.e. >= 0 and < num sample frames
  also, loop end must not be less than loop start
4-byte little endian sample frame index of loop 2 start
  must be a valid index, i.e. >= 0 and < num sample frames
4-byte little endian sample frame index of loop 2 end
  must be a valid index, i.e. >= 0 and < num sample frames
  also, loop end must not be less than loop start
4-byte little endian sample frame index of loop 3 start
  must be a valid index, i.e. >= 0 and < num sample frames
4-byte little endian sample frame index of loop 3 end
  must be a valid index, i.e. >= 0 and < num sample frames
  also, loop end must not be less than loop start
4-byte little endian sampling rate value
  should be between 100 and 65535
4-byte little endian natural frequency fractional portion
  unsigned; divide by 2^32 to get the actual fraction
4-byte little endian natural frequency integer portion
  total natural frequency should be between 0.01 and 1e6
4-byte total number of sample frames
1-byte mono/stereo flag
  1 = mono
  2 = stereo
1-byte number of bits per sample point
  should be 8 or 16
n-bytes of data for sample frames
  stereo samples have the left channel sample point preceding the right
  channel sample point.
  sample points that require more than 1 byte are stored little endian
  all sample data is stored in signed 2's complement form

Function Object Subblock Structure:

1-byte function object version number
  should be 1
2-byte little endian window x location (origin at top-left of screen)
2-byte little endian window y location
2-byte little endian window width
2-byte little endian window height
4-byte little endian object name length (positive 2s complement)
n-byte name data (line feed = 0x0a)
4-byte little endian function source text length (positive 2s complement)
n-byte function source text data (line feed = 0x0a)

Algorithmic Sample Object Subblock Format:

1-byte format version number
  should be 1
2-bytes little endian window X location (signed, origin at top-left corner)
2-bytes little endian window Y location
2-bytes little endian window width
2-bytes little endian window height
4-bytes little endian name length descriptor
n-bytes name string (line feed = 0x0a)
4-bytes little endian formula length descriptor
n-bytes formula string (line feed = 0x0a)
1-byte number of bits
  should be 8 or 16
1-byte number of channels
  1 = mono
  2 = stereo
4-bytes little endian sample origin
4-bytes little endian loop 1 start point
4-bytes little endian loop 1 end point

4-bytes little endian loop 2 start point
4-bytes little endian loop 2 end point
4-bytes little endian loop 3 start point
4-bytes little endian loop 3 end point
4-bytes little endian sampling rate
  should be between 100 and 65535
4-byte little endian natural frequency fractional portion
  unsigned; divide by 2^32 to get the actual fraction
4-byte little endian natural frequency integer portion
  total natural frequency should be between 0.01 and 1e6

Wave Table Object Subblock Format:
1-byte format version number
  should be 1
2-byte little endian window X location (signed; origin at upper left corner)
2-byte little endian window Y location
2-byte little endian window width
2-byte little endian window height
4-byte little endian wave table name length descriptor
n-byte name string (line feed = 0x0a)
4-byte little endian wave table formula length descriptor
n-byte formula string (line feed = 0x0a)
4-byte little endian large integer encoded test attack duration.
  large integer coded decimal is decimal * 1000000 with a
  range of -1999.999999 to 1999.999999
4-byte little endian large integer encoded test decay duration.
4-byte little endian test frequency fractional portion
  unsigned; divide by 2^32 to get the actual fraction
4-byte little endian test frequency integer portion
  total test frequency should be between 0.01 and 1e6
4-byte little endian test sampling rate
  should be between 100 and 65535
4-byte little endian number of tables
4-byte little endian number of frames per table
  must be an integral power or 2 between 2 and 65536
1-byte number of bits specifier
  must be 8 or 16
n-byte sample data for the wave table
  data is stored as follows: each table is stored consecutively starting
  with the table numbered 0. in each table, each sample frame is stored
  consecutively as a signed 2s complement value. 8-bit sample frames
  use 1 byte each. 16-bit sample frames use 2 bytes and are stored little
  endian.

Algorithmic Wave Table Object Subblock Format:
1-byte format version number
  should be 1
2-byte little endian window X position (signed; origin at top-left corner)
2-byte little endian window Y position
2-byte little endian window width
2-byte little endian window height
4-byte little endian name length descriptor
n-byte name string (line feed = 0x0a)
4-byte little endian formula length descriptor
n-byte formula string (line feed = 0x0a)
4-byte little endian number of frames
  should be an integral power of 2 in the interval 2..65536
4-byte little endian number of tables
1-byte number of bits
  should be either 8 or 16

Instrument Object Subblock Format:
1-byte format version number
  should be 1
2-byte little endian window X location (signed; origin at top-left of screen)
2-byte little endian window Y location
2-byte little endian window width
2-byte little endian window height
4-byte little endian name string length
n-byte name string (line feed = 0x0a)
4-byte little endian instrument definition length
n-byte instrument definition string (line feed = 0x0a)

Track Object Subblock Format:
1-byte format version number
  should be 1

2-byte little endian window X position (signed; from top-left corner of screen)
2-byte little endian window Y position
2-byte little endian window width
2-byte little endian window height
4-byte little endian track name length descriptor
n-byte track name string (line feed = 0x0a)
4-byte little endian large integer coded decimal default early/late adjust
  large integer coded decimal is decimal * 1000000 with a
  range of -1999.999999 to 1999.999999
4-byte little endian large integer coded decimal default release point 1
1-byte default release point 1 mode flag
  0 = release from start
  1 = release from end
4-byte little endian large integer coded decimal default release point 2
1-byte default release point 2 mode flag
  0 = release from start
  1 = release from end
4-byte little endian large integer coded decimal default overall loudness
4-byte little endian large integer coded decimal default stereo positioning
4-byte little endian large integer coded decimal default surround positioning
4-byte little endian large integer coded decimal default accent 1
4-byte little endian large integer coded decimal default accent 2
4-byte little endian large integer coded decimal default accent 3
4-byte little endian large integer coded decimal default accent 4
4-byte little endian large integer coded decimal default pitch disp depth adjust
1-byte default pitch displacement depth adjust mode flag
  0 = half steps
  1 = hertz
4-byte little endian large integer coded decimal default pitch disp rate adjust
4-byte little endian large integer coded decimal default pitch disp start point
1-byte default pitch displacement start point mode flag
  0 = pitch displacement point from start
  1 = pitch displacement point from end
4-byte little endian large integer coded decimal default hurry-up factor
4-byte little endian large integer coded decimal default detuning
1-byte default detuning mode flag
  0 = half steps
  1 = hertz
4-byte little endian large integer coded decimal default duration
1-byte default duration mode flag
  0 = duration adjust is multiplicative
  1 = duration adjust is additive
1-byte flag for playback inclusion
  0 = don't play track in final playback
  1 = do play track in final playback
4-byte little endian instrument name string length descriptor
n-byte instrument name string (line feed = 0x0a)
1-byte flag for channel post processing enabling
  0 = don't do channel postprocessing
  1 = do channel postprocessing
4-byte little endian postprocessing expression length descriptor
n-bytes of postprocessing stuff (line feed = 0x0a)
n-bytes of data for note vector

Note Vector Subblock Format:
1-byte format version number
  should be 1
4-byte little endian number of frames in the vector
* for each frame:
  4-byte little endian number of notes in the frame
  n-bytes of data for all of the notes (see note object format)
4-byte little endian number of records in the tie matrix
* for each tie matrix entry:
  4-byte little endian index of the source frame
  4-byte little endian index of the source note in the frame
  4-byte little endian index of the target frame
  4-byte little endian index of the target note in the frame

Note Object Subblock Format:
Each note/command has the following field:
4-byte unsigned little endian opcode field
  for a command, the high bit will be 1 and the remaining bits will
  be the opcode. for a note, the high bit will be 0.

Remainder of Note Variant Subblock:
definitions for the remaining bits in the opcode field

bits 0-3: duration integer.
0001 = 64th note
0010 = 32nd note
0011 = 16th note
0100 = 8th note
0101 = quarter note
0110 = half note
0111 = whole note
1000 = double note
1001 = quad note
all other values are not permitted
bits 4-5: division mask
00 = divide duration by 1 (no change in duration)
01 = divide duration by 3 (triplets)
10 = divide duration by 5
11 = divide duration by 7
bit 6: dotted note flag (1 = dot)
bit 7: flat modifier (1 = flat); this is used for notation only
at most 1 of the flat or sharp modifiers can be set
bit 8: sharp modifier (1 = sharp); this is used for notation only
at most 1 of the flat or sharp modifiers can be set
bit 9: rest modifier (1 = rest instead of note)
bits 10-11: release point 1 origin
01 = use default origin
10 = measure release point from start of note
11 = measure release point from end of note
other values are not permitted
bits 12-13: release point 2 origin
01 = use default origin
10 = measure release point from start of note
11 = measure release point from end of note
other values are not permitted
bit 14: release point 3 origin (1 = from start instead of end)
0 = release 3 occurs at end of note
1 = release 3 occurs at start of note
bits 15-16: pitch displacement start point origin
01 = use default origin
10 = pitch displacement start point from start of note
11 = pitch displacement start point from end of note
other values are not permitted
bits 17-18: pitch displacement depth modulation mode
01 = use default pitch modulation scale
10 = treat modulation index as half-step number
11 = treat modulation index as hertz
other values are not permitted
bits 19-20: detuning mode
01 = use default treat detuning scale
10 = treat detuning value as half-step number
11 = treat detuning value as hertz
other values are not permitted
bits 21-22: note duration adjustment scale
01 = use default duration adjustment scale
10 = add duration adjustment to the duration
11 = multiply the duration adjustment by the duration
other values are not permitted
bit 23: retrigger envelopes on tie flag
bit 24: portamento linear in hertz instead of half steps (1 = hertz)
bits 25-31: set to zero!

2-byte signed little endian pitch index
should be a value in the range 0..383. Middle C (261.6 Hertz) = 192

2-byte little endian small integer coded decimal portamento duration.
this determines how long a portamento will last, in fractions of a quarter note. it only has effect if the note is the target of a tie. a value of 0 means instantaneous, i.e. no portamento.
A small integer coded decimal is the decimal * 1000 with a range of -29.999 to 29.999

2-byte little endian small integer coded decimal early/late adjustment
this determines the displacement in time of the occurrence of the note in fractions of a quarter note.

2-byte little endian small integer coded decimal duration adjustment
this value changes the duration of the note by being added to the duration or being multiplied by the duration.

2-byte little endian small integer coded decimal release point 1 location
this determines when the release of the first sustain/loop will occur in fractions of the current note's duration. it is relative to the origin as determined by the opcode field.

2-byte little endian small integer coded decimal release point 2 location
this determines when the release of the second sustain/loop will occur.
2-byte little endian small integer coded decimal overall loudness adjustment
this factor scales the total volume output of the oscillators for this particular note. It is multiplied, so a value of 1 makes no change in loudness.
2-byte little endian small integer coded decimal stereo position adjustment.
this value adjusts where the sound will be located in stereo. -1 is the far left, 1 is the far right, and 0 is center.
2-byte little endian small integer coded decimal surround position adjustment.
this value adjusts where the sound will be located in surround sound. 1 is front and -1 is rear.
2-byte little endian small integer coded decimal accent 1 value
2-byte little endian small integer coded decimal accent 2 value
2-byte little endian small integer coded decimal accent 3 value
2-byte little endian small integer coded decimal accent 4 value
2-byte little endian fake pitch value
this value has a range of -1..383. If it is not -1, then it will be used to determine which sample a multisampled oscillator will use. If it is -1 then the actual pitch will be used to select a sample.
2-byte little endian small integer coded decimal pitch disp depth adjustment
this adjusts the maximum amplitude of the pitch displacement depth oscillator (vibrato). The value has units of either half steps or hertz depending on the setting in the opcode word.
2-byte little endian small integer coded decimal pitch disp rate adjustment
this adjusts the maximum amplitude of the pitch displacement rate oscillator. the units are periods per second.
2-byte little endian small integer coded decimal pitch disp start point adjust
this value adjusts when the pitch displacement envelopes start. the location is from start or end of note, depending on the opcode settings, and is in fractions of the current note's duration.
2-byte little endian small integer coded decimal hurry-up factor
this factor scales the total speed at which all envelopes change. this is multiplicative, so a value of 1 makes no change, and smaller values make transitions go faster.
2-byte little endian small integer coded decimal detuning value
this value is added to the pitch of the note to detune. its units are either hertz or half steps depending on the opcode word.

Remainder of Command Variant Subblock:
lower 31 bits of the command opcode:
16 = restore tempo
17 = set tempo
18 = adjust tempo
19 = sweep tempo absolute
20 = sweep tempo relative
32 = restore stereo position
33 = set stereo position
34 = adjust stereo position
35 = sweep stereo position absolute
36 = sweep stereo position relative
48 = restore volume
49 = set volume
50 = adjust volume
51 = sweep volume absolute
52 = sweep volume relative
64 = restore release point 1
65 = set release point 1
66 = adjust release point 1
67 = set release point 1 origin
68 = sweep release point 1 absolute
69 = sweep release point 1 relative
80 = restore release point 2
81 = set release point 2
82 = adjust release point 2
83 = set release point 2 origin
84 = sweep release point 2 absolute
85 = sweep release point 2 relative
96 = restore accent 1
97 = set accent 1
98 = adjust accent 1
99 = sweep accent 1 absolute
100 = sweep accent 1 relative
112 = restore accent 2
113 = set accent 2
114 = adjust accent 2

115 = sweep accent 2 absolute
116 = sweep accent 2 relative
128 = restore accent 3
129 = set accent 3
130 = adjust accent 3
131 = sweep accent 3 absolute
132 = sweep accent 3 relative
144 = restore accent 4
145 = set accent 4
146 = adjust accent 4
147 = sweep accent 4 absolute
148 = sweep accent 4 relative
160 = restore pitch displacement depth
161 = set pitch displacement depth
162 = adjust pitch displacement depth
163 = set pitch displacement depth modulation mode
164 = sweep pitch displacement depth absolute
165 = sweep pitch displacement depth relative
176 = restore pitch displacement rate
177 = set pitch displacement rate
178 = adjust pitch displacement rate
179 = sweep pitch displacement rate absolute
180 = sweep pitch displacement rate relative
192 = restore pitch displacement start point
193 = set pitch displacement start point
194 = adjust pitch displacement start point
195 = set pitch displacement start point origin
196 = sweep pitch displacement start point absolute
197 = sweep pitch displacement start point relative
208 = restore hurry-up factor
209 = set hurry-up factor
210 = adjust hurry-up factor
211 = sweep hurry-up factor absolute
212 = sweep hurry-up factor relative
224 = restore detuning
225 = set detuning
226 = adjust detuning
227 = set detuning mode
228 = sweep detuning absolute
229 = sweep detuning relative
240 = restore early/late adjust
241 = set early/late adjust
242 = adjust early/late adjust
243 = sweep early/late adjust absolute
244 = sweep early/late adjust relative
256 = restore duration adjust
257 = set duration adjust
258 = adjust duration adjust
259 = sweep duration adjust absolute
260 = sweep duration adjust relative
261 = set duration adjust mode
272 = set meter
273 = set measure number
288 = comment
304 = restore surround position
305 = set surround position
306 = adjust surround position
307 = sweep surround position absolute
308 = sweep surround position relative
320 = set transpose
321 = adjust transpose
no other values besides these are allowed!

Format of data for each command (not including the 4-byte opcode word):

16 = restore tempo
no arguments

17 = set tempo
4-byte little endian extended integer coded decimal number of beats per minute
an extended integer coded decimal is 1000 * the decimal value, with
a range of -1999999.999 to 1999999.999

18 = adjust tempo
4-byte little endian extended integer coded decimal beats per minute adjust

19 = sweep tempo absolute
4-byte little endian extended integer coded decimal target beats per minute
4-byte little endian extended integer coded decimal number of beats to reach it

20 = sweep tempo relative
4-byte little endian extended integer coded decimal target beats per minute adjust

4-byte little endian extended integer coded decimal number of beats to reach it
32 = restore stereo position
no arguments
33 = set stereo position
4-byte little endian large integer coded decimal stereo position value
-1 is left, 1 is right, 0 is center
34 = adjust stereo position
4-byte little endian large integer coded decimal stereo position adjustment
35 = sweep stereo position absolute
4-byte little endian large integer coded decimal target stereo position
4-byte little endian extended integer coded decimal number of beats to reach it
36 = sweep stereo position relative
4-byte little endian large integer coded decimal target stereo position adjust
4-byte little endian extended integer coded decimal number of beats to reach it
48 = restore volume
no arguments
49 = set volume
4-byte little endian large integer coded decimal volume level specifier
50 = adjust volume
4-byte little endian large integer coded decimal volume level adjustment
51 = sweep volume absolute
4-byte little endian large integer coded decimal target volume level
4-byte little endian extended integer coded decimal number of beats to reach it
52 = sweep volume relative
4-byte little endian large integer coded decimal target volume level adjust
4-byte little endian extended integer coded decimal number of beats to reach it
64 = restore release point 1
no arguments
65 = set release point 1
4-byte little endian large integer coded decimal release point
66 = adjust release point 1
4-byte little endian large integer coded decimal release point adjust
67 = set release point 1 origin
1-byte origin specifier
0 = from start
1 = from end
68 = sweep release point 1 absolute
4-byte little endian large integer coded decimal target release point
4-byte little endian extended integer coded decimal number of beats to reach it
69 = sweep release point 1 relative
4-byte little endian large integer coded decimal target release point adjust
4-byte little endian extended integer coded decimal number of beats to reach it
80 = restore release point 2
no arguments
81 = set release point 2
4-byte little endian large integer coded decimal release point
82 = adjust release point 2
4-byte little endian large integer coded decimal release point adjust
83 = set release point 2 origin
1-byte origin specifier
0 = from start
1 = from end
84 = sweep release point 2 absolute
4-byte little endian large integer coded decimal target release point
4-byte little endian extended integer coded decimal number of beats to reach it
85 = sweep release point 2 relative
4-byte little endian large integer coded decimal target release point adjust
4-byte little endian extended integer coded decimal number of beats to reach it
96 = restore accent 1
no arguments
97 = set accent 1
4-byte little endian large integer coded decimal accent value
98 = adjust accent 1
4-byte little endian large integer coded decimal accent adjust
99 = sweep accent 1 absolute
4-byte little endian large integer coded decimal target accent
4-byte little endian extended integer coded decimal number of beats to reach it
100 = sweep accent 1 relative
4-byte little endian large integer coded decimal target accent adjust
4-byte little endian extended integer coded decimal number of beats to reach it
112 = restore accent 2
no arguments
113 = set accent 2
4-byte little endian large integer coded decimal accent value
114 = adjust accent 2
4-byte little endian large integer coded decimal accent adjust

115 = sweep accent 2 absolute
 4-byte little endian large integer coded decimal target accent
 4-byte little endian extended integer coded decimal number of beats to reach it
116 = sweep accent 2 relative
 4-byte little endian large integer coded decimal target accent adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
128 = restore accent 3
 no arguments
129 = set accent 3
 4-byte little endian large integer coded decimal accent value
130 = adjust accent 3
 4-byte little endian large integer coded decimal accent adjust
131 = sweep accent 3 absolute
 4-byte little endian large integer coded decimal target accent
 4-byte little endian extended integer coded decimal number of beats to reach it
132 = sweep accent 3 relative
 4-byte little endian large integer coded decimal target accent adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
144 = restore accent 4
 no arguments
145 = set accent 4
 4-byte little endian large integer coded decimal accent value
146 = adjust accent 4
 4-byte little endian large integer coded decimal accent adjust
147 = sweep accent 4 absolute
 4-byte little endian large integer coded decimal target accent
 4-byte little endian extended integer coded decimal number of beats to reach it
148 = sweep accent 4 relative
 4-byte little endian large integer coded decimal target accent adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
160 = restore pitch displacement depth
 no arguments
161 = set pitch displacement depth
 4-byte little endian large integer coded decimal pitch disp depth
162 = adjust pitch displacement depth
 4-byte little endian large integer coded decimal pitch disp depth adjust
163 = set pitch displacement depth modulation mode
 1-byte mode flag
    0 = hertz
    1 = half steps
164 = sweep pitch displacement depth absolute
 4-byte little endian large integer coded decimal target pitch disp depth
 4-byte little endian extended integer coded decimal number of beats to reach it
165 = sweep pitch displacement depth relative
 4-byte little endian large integer coded decimal target pitch disp depth adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
176 = restore pitch displacement rate
 no arguments
177 = set pitch displacement rate
 4-byte little endian large integer coded decimal pitch disp rate
178 = adjust pitch displacement rate
 4-byte little endian large integer coded decimal pitch disp rate adjust
179 = sweep pitch displacement rate absolute
 4-byte little endian large integer coded decimal target pitch disp rate
 4-byte little endian extended integer coded decimal number of beats to reach it
180 = sweep pitch displacement rate relative
 4-byte little endian large integer coded decimal target pitch disp rate adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
192 = restore pitch displacement start point
 no arguments
193 = set pitch displacement start point
 4-byte little endian large integer coded decimal pitch disp start point
194 = adjust pitch displacement start point
 4-byte little endian large integer coded decimal pitch disp start point adjust
195 = set pitch displacement start point origin
 1-byte start point origin
    0 = from start
    1 = from end
196 = sweep pitch displacement start point absolute
 4-byte little endian large integer coded decimal target pitch disp start point
 4-byte little endian extended integer coded decimal number of beats to reach it
197 = sweep pitch displacement start point relative
 4-byte little endian large integer coded decimal target pitch disp start adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
208 = restore hurry-up factor
 no arguments

209 = set hurry-up factor
 4-byte little endian large integer coded decimal hurry-up factor
210 = adjust hurry-up factor
 4-byte little endian large integer coded decimal hurry-up factor adjust
211 = sweep hurry-up factor absolute
 4-byte little endian large integer coded decimal target hurry-up factor
 4-byte little endian extended integer coded decimal number of beats to reach it
212 = sweep hurry-up factor relative
 4-byte little endian large integer coded decimal target hurry-up factor adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
224 = restore detuning
 no arguments
225 = set detuning
 4-byte little endian large integer coded decimal detuning
226 = adjust detuning
 4-byte little endian large integer coded decimal detuning adjust
227 = set detuning mode
 1-byte detuning mode specifier
    0 = hertz
    1 = half steps
228 = sweep detuning absolute
 4-byte little endian large integer coded decimal target detuning
 4-byte little endian extended integer coded decimal number of beats to reach it
229 = sweep detuning relative
 4-byte little endian large integer coded decimal target detuning adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
240 = restore early/late adjust
 no arguments
241 = set early/late adjust
 4-byte little endian large integer coded decimal early/late adjust
242 = adjust early/late adjust
 4-byte little endian large integer coded decimal early/late adjust adjustment
243 = sweep early/late adjust absolute
 4-byte little endian large integer coded decimal target early/late adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
244 = sweep early/late adjust relative
 4-byte little endian large integer coded decimal target early/late adjust adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
256 = restore duration adjust
 no arguments
257 = set duration adjust
 4-byte little endian large integer coded decimal duration adjust
258 = adjust duration adjust
 4-byte little endian large integer coded decimal duration adjust adjustment
259 = sweep duration adjust absolute
 4-byte little endian large integer coded decimal target duration adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
260 = sweep duration adjust relative
 4-byte little endian large integer coded decimal target duration adjust adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
261 = set duration adjust mode
 1 byte mode flag
    0 = multiplicative
    1 = additive
272 = set meter
 4-byte little endian numerator
 4-byte little endian denominator
273 = set measure number
 4-byte little endian new measure number
288 = comment
 4-byte little endian length of comment text string
 n-byte comment text string (line feed = 0x0a)
304 = restore surround position
 no arguments
305 = set surround position
 4-byte little endian large integer coded decimal surround position value
    1 is front, 0 is middle, -1 is rear
306 = adjust surround position
 4-byte little endian large integer coded decimal surround position adjustment
307 = sweep surround position absolute
 4-byte little endian large integer coded decimal target surround position
 4-byte little endian extended integer coded decimal number of beats to reach it
308 = sweep surround position relative
 4-byte little endian large integer coded decimal target surround position adjust
 4-byte little endian extended integer coded decimal number of beats to reach it
320 = set transpose

4-byte signed little endian half-step change (<0 decreases pitch)
321 = adjust transpose
4-byte signed little endian transpose adjustment

Background Display Subblock Format:
4-byte little endian number of tracks in the background of this track
for each of those:
4-byte index of the track to be in the background
the index is respective to the order in which the tracks were loaded
from the file, so 0 is the first track, 1 is the second, and so on.