Confirmation of Approval
for
MIDI Standard
MMA/AMEI

Date of issue:___3/01/99
Originated by:___MMA_____

Reference TSBB Item #:_____141_____   TSBB Volume #: 22 (revised)

Title: ___MIDI Tuning Bank and Dump Extensions
CA#: ΩΩΩ___

Related item(s): SCALE/OCTAVE TUNING, MIDI Tuning Standard

Abstract:

This proposal replicates the existing Tuning Dump Messages with the addition of a bank select byte. It also provides Tuning Dump Messages for devices supporting Scale/Octave Tuning

Background:

In the MIDI Spec V4.2, document version 96.1, page 50, a Tuning Bank Select Registered Parameter Number (RPN) is defined for choosing among banks of tuning programs. However, no way to store a tuning message in a specific bank is defined. The bulk tuning request and response messages should have a bank number in addition to the tuning program number.

Details:

The first three messages are based on existing MIDI Tuning messages, with the addition of the bank select byte. The next three messages are new messages to support Scale/Octave tables (see Scale/Octave Tuning) and to provide a non real-time version of the Single Note Tuning message.

The intent of a non-real time message is that it should specifically be sent as a setup message. If it is NOT sent as a setup message (that is, if it is sent during performance), it is assumed that the message will be ignored for notes that are already sounding. However, certain Recommended Practices, (RPs), may implement this differently. Manufacturers should therefore consult the specific RP for the required response and should document the device's response in the user manual.

Message #1
BULK TUNING DUMP REQUEST (BANK)

```
F0 7E <device ID> 08 03 bb tt F7

        F0 7E        Universal Non-Real Time SysEx header
        <device ID>  ID of target device (7F=all devices)
        08           sub-ID#1 = "MIDI tuning standard"
        03           sub-ID#2 = "BULK TUNING DUMP REQUEST (BANK)"
        bb           bank: 0-127
                     (described as 1-128 in MIDI Tuning Specification)
        tt           tuning preset number: 0-127
        F7           EOX
```

This is identical to the current BULK TUNING DUMP REQUEST except for the addition of the bank select byte (bb). The receiving device will send back a Tuning Dump message which will vary depending on which tables, (key-based or scale/octave), are supported by the instrument.

Message #2
KEY-BASED TUNING DUMP

```
FO 7E <device ID> 08 04 bb tt <tuning name> [xx yy zz] ... chksum F7
```

This is identical to the current BULK TUNING DUMP except for the addition of the bank select byte (bb). The message was renamed KEY-BASED TUNING DUMP to differentiate between it and the new scale/octave tuning messages. (See below)

```
FO 7E Universal Non-Real Time SysEx header
<device ID>          ID of target device (7F=all devices)
08                   sub-ID#1 = "MIDI tuning standard"
04                   sub-ID#2 = "KEY-BASED TUNING DUMP"
bb                   bank: 0-127 (described as 1-128 in MIDI Tuning
                     Specification)
tt                   tuning preset number: 0-127
<tuning name>        16 ASCII characters
[xx yy zz]           frequency data for one note (repeated 128 times)
checksum             see "Checksum calculation", below
F7 EOX
```

Message #3
SINGLE NOTE TUNING CHANGE (REAL-TIME) (BANK)

```
FO 7F <device ID> 08 07 bb tt ll [kk xx yy zz] ... F7
```

This is identical to the current SINGLE NOTE TUNING CHANGE (REAL-TIME) except for the addition of the bank select byte (bb). This message WILL affect currently sounding notes.

```
FO 7F                Universal Real Time SysEx header
<device ID>          ID of target device (7F=all devices)
08                   sub-ID#1 = "MIDI tuning standard"
07                   sub-ID#2 = "SINGLE NOTE TUNING CHANGE (REAL-
                     TIME)(BANK)"
bb                   bank: 0-127 (described as 1-128 in MIDI Tuning
                     Specification)
tt                   tuning preset number: 0-127
ll                   number of changes (1 change = 1 set of [kk xx yy
                     zz])
[kk                  MIDI key number
xx yy zz]            frequency data for that key (repeated 'll'
                     number of times)
F7                   EOX
```

Message #4
SINGLE NOTE TUNING CHANGE (NON REAL-TIME) (BANK)

```
FO 7E <device ID> 08 07 bb tt ll [kk xx yy zz] ... F7
```

This is identical to the current SINGLE NOTE TUNING CHANGE (REAL-TIME) except for the addition of the bank select byte (bb) and the change to a NON REAL -TIME header. This message allows the sender to specify a new tuning change that will NOT update the currently sounding notes.

FO 7E	Universal Non-Real Time SysEx header
<device ID>	ID of target device (7F=all devices)
08	sub-ID#1 = "MIDI tuning standard"
07 s	sub-ID#2 = "SINGLE NOTE TUNING CHANGE (NON REAL-TIME)(BANK)"
bb	bank: 0-127 (described as 1-128 in MIDI Tuning Spec)
tt	tuning preset number: 0-127
ll	number of changes (1 change = 1 set of [kk xx yy zz])
[kk	MIDI key number
xx yy zz]	frequency data for that key (repeated 'll' number of times)
F7	EOX

Message #5
SCALE/OCTAVE TUNING DUMP, 1 byte format

```
FO 7E <device ID> 08 05 bb tt <tuning name> [xx] ... chksum F7
```

FO 7E	Universal Non-Real Time SysEx header
<device ID>	ID of target device (7F=all devices)
08	sub-ID#1 = "MIDI tuning standard"
05	sub-ID#2 = "SCALE/OCTAVE TUNING DUMP, 1 byte format"
bb	bank: 0-127 (described as 1-128 in MIDI Tuning Specification)
tt	tuning preset number: 0-127
<tuning name>	16 ASCII characters
[xx]	frequency data for C,C#,... B (12 bytes total) 00H means -64 Cent 40H means +/- 0 Cent 7FH means +63 Cent
chksum	see "Checksum calculation", below
F7	EOX

Message #6
SCALE/OCTAVE TUNING DUMP, 2 byte format

```
FO 7E <device ID> 08 06 bb tt <tuning name> [xx yy] ... chksum F7
```

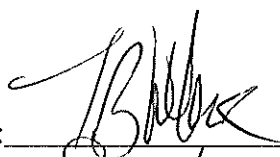|	
FO 7E	Universal Non-Real Time SysEx header
<device ID>	ID of target device (7F=all devices)
08	sub-ID#1 = "MIDI tuning standard"
06	sub-ID#2 = "SCALE/OCTAVE TUNING DUMP, 2 byte format"
bb	bank: 0-127
	(described as 1-128 in MIDI Tuning Specification)
tt	tuning preset number: 0-127
<tuning name>	16 ASCII characters
[xx yy]	frequency data for C,C#,... B (24 bytes total)
	00H 00H means -100 cents (8,192 steps of .012207 cents)
	40H 00H means 0 cents (equal temperament)
	7FH 7FH means +100 cents (8,191 steps of .012207 cents)
chksum	see "Checksum calculation", below
F7	EOX

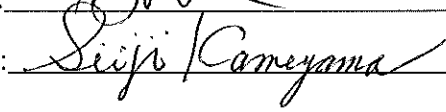The minimum and maximum offsets are approximately 100 cents.

Checksum calculation:

The checksum field, where present, is calculated by successively XOR'ing the bytes in the message, excluding the FO, F7, and the checksum field. The resulting value is then AND'ed with 7F, to create a 7 bit value.

Notes:

GM Level 2 will not support tuning presets, but it will support on-the-fly retunings (proposal #146).

Approved by MMA:_____   Date: 5/12/99

Approved by AMEI:_____   Date: 5/7/99